

محاضرات مادة

( انظمة تشغيل )

**( OPERATING SYSTEMS )**

قسم الالكترونيك

فرع تقنيات الحاسوب

المرحلة الثانية

اعداد

مدرسة المادة

بشرى خضير عباس

**(نظام التشغيل )**

**(( مفاهيم عامة ))**

## نظام التشغيل (operating system)

هو عبارة عن مجموعة من البرمجيات الأساسية التي تقوم بإدارة جهاز الحاسوب والتحكم في كافة البرامج والتطبيقات. تسهل هذه البرمجيات على المستخدم الاستفادة من الأجهزة التي يتكون منها الحاسوب والملحقات التابعة له.

يعد نظام التشغيل أهم برمجيات الحاسوب بل هو عصب الحياة بالنسبة للحاسوب وتختلف مهامه باختلاف الحواسيب وانواعها .

### مراحل تطور نظام التشغيل:

**في أربعينيات:** القرن الماضي كانت الأجهزة بدائية وبسيطة بحيث لا تتطلب أنظمة تشغيل لعملها لأنها تنفذ برنامج واحد وعند انتهائه يبدأ برنامج ثاني وهكذا. في هذه الفترة كان على المستخدم كتابة جميع العمليات بلغة الآلة بحيث كان المستخدم هو المبرمج بالإضافة إلى كتابته عمليات الإدخال والإخراج بالتفصيل. وبعد زمن جمعت جميع عمليات الإدخال والإخراج التي كانت في السابق تشكل صعوبة في بناء البرامج ، جمعت هذه العمليات لتكوين مكتبة عمليات ادخال واخراج سميت :

(input/ output control system) : ( iocs ) وعندما يرغب المبرمج في اجراء عملية ادخال او اخراج ما عليه الا اجراء نداء لاحدى الدوال التي تحتويها المكتبة لتقوم بالغرض ، هذه المكتبة كانت النواة التي منها تطور مبدأ نظام التشغيل حيث كانت هذه المكتبة مخزونة في الآلة لتسهيل عملها والعمل عليها . في هذه الانظمة قضى المستخدم وقتا طويلا في تحويل الجهاز من عملية الى اخرى ، اذ عند انتهاء احد البرامج كان على المستخدم ازالة جميع الاشرطة المغناطيسية المحتوية على بيانات البرنامج ، وازالة البطاقات التي تحتوي على الاوامر ليضع بدلها الاشرطة التي تحتوي على البرنامج او العملية القادمة . وكحل لهذه المشكلة صمم نظام ميكانيكي للتحويل من برنامج لاخر سنة ( 1956 ) . لاقت هذه الطريقة نجاحا كبيرا وكانت بداية لانظمة المعالجة بالدفعات .

### **في الخمسينيات: ظهر نظام المعالجة الحزمية ( batch processing )**

ويسمى نظام المعالجة بالدفعات. والمعالجة بالدفعات تعني تنفيذ سلسلة من الوظائف على كمبيوتر بدون تدخل يدوي. سميت بهذه التسمية لان بيانات المدخلات يتم جمعها في دفعات من الملفات وتتم معالجتها في دفعات من خلال البرنامج.

نظام التشغيل كان بسيطا في هذه الفترة . استعمل هذا النظام للتحكم بالكمبيوترات المركزية الضخمة ( Main frame ). كانت الكمبيوترات في ذلك الوقت اجهزة ضخمة تستخدم البطاقات المثقبة للإدخال ، والطابعة الخطية للإخراج ، والاشرطة الممغنطة كوسط تخزين ثانوي ، والبرنامج الذي ينفذ يسمى مهمة ( job ) .

## آلية عمل النظام :

تجمع جميع الوظائف في مجموعة واحدة من البطاقات يتعرف عليها الجهاز، ويفصل بين كل وظيفة وأخرى بطاقة تحكم . تتحكم الاجهزة بهذه البطاقات المحتوية على الوظائف عن طريق لغة تدعى : ( jcl ) اي : (job control language) ، فعند انتهاء بطاقة وظيفة يقرأ الجهاز بطاقة التحكم التي تليها المحتوية على معلومات تخص الوظيفة القادمة .

**في الستينات:** تطورت أنظمة التشغيل بتطور الحاسوب وتطور وحداته وأجهزته وذاكرته. وظهر نظام تشغيل قادر على التعامل مع أكثر من برنامج في نفس الوقت سمي : نظام التشغيل متعدد البرامج (multi programming). وهذه الطريقة توفر اكبر استخدام للمعالج ولمصادر الإدخال والإخراج.

المشكلة الوحيدة التي واجهتها الأنظمة في ذلك الوقت هي محدودية قدرة الأجهزة على تخزين الوظائف التي ستعمل عليها في نفس الوقت . كحل للمشكلة ظهر نظام المشاركة الزمنية (time sharing system) أي اشراك أكثر من مستخدم بنفس الذاكرة بحيث يعطى كل مستخدم شريحة زمنية من الوقت الكلي للحاسوب بحيث ينفذ عمل كل مستخدم بمقدار تلك الشريحة الزمنية المعطاة له ، ومن ثم يعطى مرة أخرى كل مشترك لم ينفذ عمله شريحة زمنية اضافية ، وهكذا حتى تنتهي جميع الاعمال المرتبطة بالذاكرة الاساسية .  
مثال : عندما يستخدم برنامج ما جهاز الادخال أو الاخراج تكون وحدة المعالجة المركزية ( cpu ) متاحة فيمكن استخدامها بواسطة برنامج آخر .

**ملاحظة :** شريحة الوقت المخصصة للعمل على كل وظيفة تم اختيارها بدقة حيث انها مدة صغيرة جدا في ادراك البشر لكي يتمكن المعالج من انجاز عملا كثيرا بها ، وهكذا يظن كل مستخدم بان المعالج مكرس لانجاز برنامجه بينما هو في الواقع يعمل على اكثر من برنامج في نفس الوقت .

**في السبعينيات :** تطور مبدأ تسويق البرامج ونظم التشغيل. وبانتشار الشبكات احتاجت نظم التشغيل في تلك الفترة إلى تطوير إمكانياتها الشبكية وكذلك تطوير نظم الأمن والحماية فيها ، فكان الهدف إيجاد نظام تشغيل آمن ومحمي ضد الفيروسات والمتدخلين.

تطور نظام التشغيل ( unix ) في هذه الفترة ، فبعد كتابته بلغة الالة كتب بلغة C وانتشر بسبب كونه اول نظام تشغيل كتب بلغة ذات مستوى عالي .

في الثمانينيات : ثورة الحاسوب الشخصي . إذ تطورت الحاسبة فأصبح لكل حاسبة نظام تشغيل متطور خاص بها .

ولازالت الحاسبات في تطور شأنها شأن اي جهاز الكتروني ، وانظمة التشغيل تتطور تبعاً لها حسب التطبيق المستخدمة فيه الحاسبة .

### الوظائف الأساسية لنظام التشغيل

1. تنظيم مسار البيانات وملفات المستخدم على العديد من وسائط الخزن ، كالقرص الصلب والقرص المضغوط . وكل نظام تشغيل يعتمد على ملفات خاص به . مثلاً تعتمد معظم أنظمة تشغيل مايكروسوفت ويندوز على نظام (NTFS).
2. تنظيم البرامج المحملة على الحاسوب وتنظيم قطع الأجهزة المتصلة به كالشاشة والطابعة ولوحة المفاتيح... الخ.
3. إدارة الذاكرة الرئيسية ووحدات الإدخال والإخراج وإدارة وحدة المعالجة المركزية ووحدات التخزين الثانوي.
4. اكتشاف الأعطال ومعالجة أخطاء قطع الأجهزة والبرامج بتشغيل سلسلة من البرامجيات الخاصة باكتشاف الأعطال.
5. تنظيم أسلوب المواجهة مع مستخدم الحاسوب، ومن خلال ذلك يتمكن مستخدم الحاسوب من متابعة البرامجيات التي يتم تنفيذها والاطلاع على الملاحظات التي يظهرها الحاسوب على الشاشة ، وكذلك إصدار الأوامر لتوجيه الحاسوب للقيام بالمهام المناسبة.

**1. نظام تشغيل الوقت الحقيقي (Real time operating system)**

وهي نظم تشغيل سريعة صممت للتحكم في الآلات والأدوات العلمية والنظم الصناعية. تتحكم هذه النظم بالحواسيب التي تتفاعل مع البيئة لتنفيذ العمل بحيث تكون عملية الإدخال والمعالجة في نفس اللحظة مثل أجهزة تخطيط القلب. وتصمم هذه الأنظمة للقيام بمهام معدودة في فترة زمنية محددة مثل: آلات تحكم ، أجهزة قياس ، أنظمة صناعية.

**2. مستخدم واحد ومهمة واحدة (Single user- single task)**

هذا النوع من الأنظمة لا يسمح لأكثر من مستخدم واحد بتشغيل أكثر من تطبيق في نفس الوقت . وفي حالة احتياجه لتشغيل تطبيق آخر لابد من إنهاء التطبيق السابق ليتمكن من العمل على التطبيق الحالي مثل نظام (MS-DOS) .

**3. مستخدم واحد وعدة مهام (Single user-multi tasking)**

هذا النوع هو الأكثر استخداما على حاسبات سطح المكتب والمحمول . يسمح هذا النوع من الأنظمة لمستخدم واحد فقط بتنفيذ أكثر من برنامج في نفس الوقت، مثلا كتابة تقرير في برنامج محرر نصوص اثناء تحميل ملف من الانترنت وطباعة نص بريد الكتروني في نفس الوقت . من أمثلة هذا النوع ماكنتوش ومايكروسوفت ويندوز (98) .

**4. متعدد المستخدمين ومهمة واحدة (Multi user –single task)**

يسمح هذا النوع من النظم أن ينفذ كل شخص ضمن مجموعة من الأشخاص برنامج في وقت محدد ، ويزود كل مستخدم بحاسوب (محطة طرفية) يتصل بحاسوب مركزي . ويسمى نظام المشاركة الزمنية لان نظام التشغيل يأمر الحاسوب بالانتقال بسرعة كبيرة بين المستخدمين بعد إعطائهم فترات زمنية ثابتة لاستخدام الـ (cpu) . ومن أمثله (window NT) .

**5. متعدد المستخدمين ومتعدد المهام (Multi user-multi tasking)**

في هذا النوع تظهر فكرة المشاركة بالوقت مع فكرة تعدد المهمات . يتطلب هذا النوع حاسبات قوية في مواصفاتها . وهو من أقوى الأنظمة ويتميز بسرعة عالية في التشغيل والتنفيذ.

فكرته مجموعة مستخدمين يمكن أن ينفذوا أكثر من عمل مرة واحدة وأي خطأ يرتكبه مستخدم ما لا يؤثر على باقي المستخدمين ولا يعطل أعمال الحاسوب . ومن أمثلته: (IBMVM,unix 0.8,unix) .

### 6. نظم تشغيل الشبكة

نظم تشغيل الشبكة تمكن عدة حواسيب منفصلة أن تتصل مع بعضها البعض وتتشارك في الموارد والبرامجيات وارسال واستقبال الرسائل عبر الشبكة . من أمثلته ( distributed unix ) .

### 7. نظم تشغيل الذاكرة الافتراضية

الذاكرة الافتراضية هي جزء من القرص الصلب يقوم نظام التشغيل بتحويله للعمل وكأنه ذاكرة . تعمل هذه الذاكرة مع الذاكرة الرئيسية كوحدة واحدة. وبذلك توسيع حجم الذاكرة بشكل كبير. ومن أمثلته: (IBMVN,UNIX) .

### 8. الأنظمة المتوازية (parallel systems)

هذا النوع ينفذ تعليمات عديدة بشكل متوازي في حاسوب واحد يمتلك وحدات معالجة مركزية عديدة. أهم ميزة في هذه الأنظمة هي السرعة بسبب وجود أكثر من (cpu) . ومن أمثلته: (N-cube,sun,o.s.) .

### 9. الأنظمة الموزعة (distributed systems)

بانتشار الشبكات أصبحت نظم التشغيل أكثر تعقيدا فالمهمات التي كانت تؤدي على حاسوب واحد أصبحت مشتركة بين حواسيب تبعد عن بعضها آلاف الأميال ، إضافة إلى إن المصادر أصبحت موزعة بين الحواسيب .

وهذا النوع يجمع المزايا السابقة لنظم التشغيل مع المهام الجديدة كتوفير الحماية والأمان.

## خدمات نظام التشغيل: وهي على نوعين:

أولاً: خدمات نظام التشغيل التي تساعد المستخدم بشكل مباشر.

ثانياً: خدمات نظام التشغيل عبر تقاسم الموارد (Resource sharing).

أولاً: خدمات نظام التشغيل التي تساعد المستخدم بشكل مباشر:

### 1. واجهة المستخدم (User interface):

جميع أنظمة التشغيل تحتوي على واجهة للمستخدم. وتأخذ هذه الواجهة أكثر من شكل.

ملاحظة: بعض أنظمة التشغيل مزودة باثنين أو أكثر من الواجهات المختلفة.

### 2. تنفيذ البرامج (Programs execution)

يجب أن تكون قدرة كافية لنظام التشغيل لتحميل البرامج في الذاكرة وتنفيذها . ويجب أن ينتهي أي تطبيق بطريقة عادية أو غير عادية عند وجود أخطاء.

### 3. عمليات الإدخال والإخراج (I/O Operation)

أي برنامج يتم تطبيقه قد يكون بحاجة إلى عمليات إدخال وإخراج بحيث يقوم بطلب ملف معين أو طلب أجهزة إدخال وإخراج ، ونظام التشغيل هو المسؤول عن القيام بذلك.

### 4. تشكيل نظام ملفات (File system manipulation)

نظام الملفات أهمية خاصة في نظم التشغيل لان البرامج تقوم بعمليات كثيرة على الملفات كقراءة وكتابة الملفات أو تكوين وحذف الملفات والأدلة أو البحث عن ملف معين وغيرها من العمليات التي تتم على الملفات .

### 5. الاتصالات (Communications)

تحتاج العمليات في بعض الأحيان الاتصال مع بعضها لتبادل المعلومات والبيانات ومشاركتها فيما بينها . وهذا الاتصال قد يكون على نفس الحاسوب أو على حاسبات مختلفة عبر شبكة.

### 6. كشف الأخطاء (error detecting)

يقوم نظام التشغيل بمراقبة النظام بشكل مستمر لاكتشاف الأخطاء التي قد تحدث ويقوم بالإجراءات المناسبة لتصحيحها عند حدوثها. لان خطأ واحد في جزء من النظام قد يسبب عطل كامل في النظام .



ثانياً: خدمات نظام التشغيل في تقاسم الموارد:

### 1. تخصيص الموارد (Resources Allocation)

عند وجود أكثر من مستخدم أو أكثر من عمل يتم تنفيذه بنفس الوقت يجب أن يتم تخصيص الموارد لكل منهم.

### 2. المحاسبة (Accounting)

تستخدم هذه الخدمة من أجل تتبع المستخدمين ومعرفة أنواع الموارد المستخدمة من قبل كل مستخدم.

### 3. الحماية والأمن (Protection and security)

يطلب الأشخاص الذين يمتلكون معلومات في أجهزة موصولة بشبكة أو في جهاز آخر يستخدمه عدد من المستخدمين يطلبون ضمان حماية معلوماتهم وعدم تداخل العمليات التي تتم في نفس الوقت.

## واجهة مستخدم نظام التشغيل (User operating system interface)

وهي الواجهة المرئية لمستخدمي النظام. وهي عبارة عن قشرة (shell) أو غلاف لنظام التشغيل . فالقشرة هي برنامج خدمة يعمل في الطبقة العليا من النظام يتيح للمستخدمين إدخال الأوامر والوصول إلى نظام التشغيل.

### من مهام (فوائد) واجهة المستخدم:-

1. تزود المستخدمين بواجهة يتم التعامل من خلالها.
2. تمكن من الوصول إلى خدمات النواة ( kernel ).
3. تمكن من تشغيل التطبيقات أو البرامج.
4. إمكانية استعراض محتويات الأدلة من خلالها.

### أنواع وأشكال واجهات المستخدم

#### 1. واجهة الأوامر النصية (Commando line interface) أي (CLI)

تعتبر من أقدم الواجهات التي تعتمد على كتابة الأوامر بواسطة لوحة المفاتيح ثم يتلقى المستخدم إجابة من النظام . هذه الواجهة لا تدعم التنفيذ المتعدد المهام . ومن الأنظمة التي تستخدمها نظام (MS-DOS) بإصداراته المختلفة.

#### 2. واجهة المستخدم الرسومية (Graphical user interface) (GUI)

وهي عبارة عن واجهة توفر صور وأيقونات تسهل عمل المستخدم وتعتبر من أكثر الواجهات استخداما . من الأنظمة التي تستخدم هذه الواجهات ويندوز بإصداراته المختلفة ولينكس ويونكس وغيرها.

#### 3. الواجهة الرسومية القابلة للتكبير (Zommable user interface) (ZUI)

وهي تختلف عن الواجهة العادية حيث إنها لا تستخدم النوافذ ، وتكون جميع التطبيقات قيد التشغيل ولكنها تكون مصغرة على سطح المكتب ، وبمجرد النقر على أيقوناتها يقوم نظام التشغيل بتكبيرها وعند الانتهاء يقوم بتصغيرها .

ومن التطبيقات التي تستخدم هذا النوع من الواجهات Google maps, Google earth لمعرفة المواقع .

## نداءات النظام (System calls)

وهي ميكانيكية تستخدمها برامج التطبيقات للحصول على خدمة يقوم بها نظام التشغيل. وهي قسمان قسم يتعامل مع المعالج وقسم يتعامل مع الملفات .

- متى تحدث النداءات؟

تحدث في وقت معالجة برامج التشغيل في الذاكرة حيث تحتاج إلى خدمات نظام التشغيل مثل استخدام الأجهزة الملحقة بالنظام كبطاقة الشبكة أو بطاقة الصوت أو الاتصالات بين البرامج التطبيقية أو البرامج التي تعاني من مشكلة معينة .

- كيف تستخدم نداءات النظام؟

عندما تستخدم نداءات النظام يتم تحديد اسم الدالة المطلوب مناداتها. وعند الحاجة إلى معلومات إضافية يتم إرسال هذه البيانات الإضافية عن طريق معاملات (Parameters) .

### مثال:

عند طباعة ورقة تحمل مستند معين فهذا أمر طبع Print فيتم استدعاء الدوال الجاهزة المخزونة في نظام التشغيل والمخصصة لعملية الطباعة مع معاملات الخاصة بها ، وهكذا في كل عمليات الإدخال والإخراج مثلا . فعملية استدعاء هذه الدوال تسمى (System calls).

### أنواع نداءات نظام التشغيل

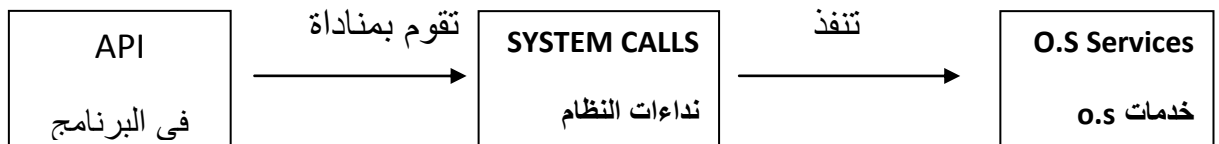
1. أعمال الملفات : مثلا : خلق ملف، حذف ملف، أو فتح ملف أو قراءة ملف أو كتابة على ملف.
2. إدارة الأجهزة : تشمل طلب تحرير أو طلب قراءة أو طلب كتابة .
3. صيانة المعلومات : تشمل طلب معلومات كمعرفة الوقت والتاريخ .
4. التواصل : تشمل خلق روابط ، أو حذف روابط ، إرسال واستقبال الرسائل .
5. التحكم في العمليات.

### الأوامر البرمجية (API) (Application program interface)

لكل نظام تشغيل مجموعة من الأوامر البرمجية التي تقوم بمناداة نداءات النظام. وليس شرط أن يكون لكل أمر برمجي نداء واحد للنظام فقد يتطلب المئات من نداءات النظام.

ومن أمثلة الأوامر البرمجية: print, Read, Open, ....

والمخطط التالي يوضح علاقة الاوامر البرمجية مع نداءات النظام :



## برامج نظام التشغيل ( system programs ) :

وهي مجموعة برامج تسهل التخاطب بين نظام التشغيل والبرامج المطورة من قبل المستخدمين ومطوري البرامج ، واكثر المستخدمين يتعاملون مع نظام التشغيل عن طريق برامج النظام وليس عن طريق الاتصال المباشر بنظام التشغيل .

### انواع برامج نظام التشغيل :

- 1 – برامج ادارة الملفات : وهي المسؤولة عن العمليات التي تتم على الملفات (خلق ، حذف ، نسخ ،...) .
- 2 – برامج حالة النظام : وهي برامج تسأل عن معلومات النظام ( الوقت ، التاريخ ، حجم الذاكرة ، عدد المستخدمين،..) .
- 3 – تعديل الملفات : وهي عبارة عن مجموعة من محررات النصوص لعمل تغييرات في محتويات الملفات
- 4 – دعم ملفات البرمجة : وهي المسؤولة عن التجميع في برامج لغات البرمجة .
- 5 – تنفيذ وتحميل البرامج : وهي المسؤولة عن تنفيذ البرامج بعد تحميلها .
- 6 – الاتصالات : وهي البرامج المسؤولة عن التواصل بين العمليات او بين المستخدمين او بين اجهزة واخرى .

### كيف يعمل نظام تشغيل الحاسوب (آلية عمل نظام التشغيل ) :

عند تشغيل الحاسوب فان نظام التشغيل يمر بعدة خطوات حتى يصبح جاهزا ليستقبل اوامر المستخدم . ويمكن تلخيص هذه الخطوات كما يلي :

- 1 – قراءة وتنفيذ التعليمات والاورامر من ذاكرة القراءة الثابتة ( rom ) .
- 2 – فحص وحدات الحاسوب للتأكد من سلامتها .
- 3 – تحميل نظام التشغيل من الاقراص ( الثابتة او المتحركة ) .
- 4 – استلام اوامر مستخدم الجهاز .
- 5 – تحميل البرامجيات التطبيقية وتنفيذ تعليماتها .
- 6 – العودة الى نظام التشغيل وانتظار اوامر المستخدم وتكرار الخطوات السابقة .

## تركيب نظام التشغيل

هناك عدة طرق لبناء وتركيب نظم التشغيل وهي:

### أولا : التركيب البسيط (Monolithic)

يكون في هذا النوع نظام التشغيل في مستوى واحد أو في مستويين.

الميزة الرئيسية لهذا النوع: تكلفة الإجراءات الداخلية في النظام تكون منخفضة لأنها جميعا تقع على نفس المستوى.

#### العيوب:

1. صعوبة الفهم
2. صعوبة التعديل
3. صعوبة الصيانة
4. غير موثوق فيه

### ثانيا : تركيب الطبقات (Layered)

أي إن نظام التشغيل مقسم لطبقات (مستويات) بحيث يكون كل جزء من النظام في طبقة مستقلة ، بحيث ان الطبقة Zero (Layer o) مخصصة للمكونات المادية (H/W) والطبقة N (Layer N) مخصصة لواجهة المستخدم .

الميزة الرئيسية: وجود الطبقات فيه أدى إلى تسهيل عملية الصيانة.

العيوب: المشكلة تكمن في عملية ترتيب الطبقات ، فلا توجد طريقة واضحة للترتيب.

### ثالثا : تركيب النواة (Microkernal)

#### نواة نظام التشغيل:

هي المستوى الأدنى من أي نظام تشغيل . وهي الطبقة الأولى من البرمجيات التي يتم تحميلها في الذاكرة عند إقلاع النظام أو بدء التشغيل، توفر النواة إمكانية الوصول إلى الخدمات المركزية الشائعة لكل برامج النظام والتطبيقات . هذه الخدمات تشمل جدولة المهام ، إدارة الذاكرة، الوصول للأجهزة المادية وغيرها.

#### تعريف آخر:

هي قلب كل نظام تشغيل للحاسوب حيث تقوم بدور حلقة الوصل بين عتاد الحاسوب وبرمجياته وتقوم أيضا بعملية التحكم بمصادر الجهاز كحلقة وصل بين العتاد والبرمجيات . بسبب طبيعتها الحساسة يتم تحميلها في منطقة محمية من الذاكرة .

تكون نواة النظام في هذا التركيب صغيرة جدا ولا يوضع بداخلها سوى الوظائف الأساسية .  
أما الوظائف الأخرى فتوضع في مساحة المستخدم، ويكون الاتصال بين مساحة المستخدم والنواة عن طريق الرسائل العابرة (message passing).

#### الميزات:

1. من السهل توسيع (تمدد) النظام.

2. النظام أكثر ثقة وأكثر أمنا.

العيوب: الاتصال بين مساحة المستخدم ونواة النظام عملية مكلفة.

#### رابعا: تركيب الوحدات (Modules-Based)

معظم أنظمة التشغيل الحديثة مبنية بهذه الطريقة، حيث تكون النواة الأساسية في المركز وبقية الوظائف تتفرع منها وهي مشابهة لتركيب الطبقات ولكن أكثر مرونة وأكثر كفاءة.

ادارة وحدة المعالجة المركزية

( CPU MANAGEMENT )

( CENTRAL PROCESSING UNIT MANAGEMENT )

• **برمجيات إدارة أـ (cpu):**

1. **مجدول الأعمال والبرامج (Job scheduler)**
2. **مجدول المعالجات والعمليات (process scheduler)**
3. **ضابط التحكم ( traffic controller ) .**

**وظائف برمجيات إدارة أـ (cpu):**

1. **استطلاع الذاكرة المساندة وذلك لاختيار الأعمال المستعدة للتنفيذ ويقوم بهذه المهمة مجدول الأعمال الذي يتصل ببرمجيات إدارة المعلومات لتمرير البرامج أو الملفات المستعدة للتنفيذ إلى الذاكرة الرئيسية حيث تتم تقسيم هذه البرامج والملفات إلى خطوات عمل مؤلفة من عمليات يعمل مجدول الأعمال على جدولتها وتنظيمها من أجل تنفيذها لاحقاً.**
2. **متابعة المعالج والعمليات أثناء تنفيذها وجدولة عملية التنفيذ ويقوم بهذه المهمة مجدول العمليات.**
3. **تحقيق التزامن بين المعالج والعمليات وتحديد حالة المعالج ويقوم بهذه المهمة ضابط التحكم ( traffic controller ) .**
4. **رسم سياسة متابعة العمليات والمعالجات وتحديد طريقة ربط المعالج بالعملية وفي الزمن المحدد ولفترة محددة ويتولى تنفيذ هذه المهمة مجدول العمليات.**
5. **ربط المعالج بالعملية لفترة زمنية محددة.**
6. **تحرير المعالج من عملية الربط وذلك عند الانتهاء من تنفيذ عملية معينة وترحيل العملية إلى الذاكرة الرئيسية حتى يستطيع المعالج الارتباط بعملية أخرى لتنفيذها.**

**جدولة الأعمال :**

تعني وضع الأعمال في قائمة الاستعداد بحيث ترتب هذه الأعمال في الجدول حسب أولويات محددة وتتم متابعة العمل من خلال مسجل خاص يسمى مسجل تعريف العمل (Job control block register) ويحتوي هذا المسجل على الحقول التالية:

1. **اسم العمل**
2. **الحالة الحالية للعمل**
3. **أولوية العمل**
4. **الزمن اللازم أو المتبقي للتنفيذ**



## ملاحظة 1:

عندما تصبح الأعمال قابلة للتنفيذ يقوم مجدول الأعمال باستطلاع مسجل التعريف لكافة الأعمال للتعرف على حالة كل عمل ويختار الأعمال التي في حالة استعداد (Ready) ويكون من هذه الأعمال قائمة الاستعداد بحيث تترتب هذه القائمة باستخدام جدول محددة.

## ملاحظة 2:

إذا كانت للحاسوب قدرة على تعدد البرامج فانه توجد هناك أعمال في الذاكرة المساندة مستعدة للتنفيذ.

## أسباب جدولة الأعمال:

1. عدد المصادر المادية في الحاسوب محدودة لذا لا بد من جدولة الأعمال وذلك لتحقيق عملية الاستخدام الأمثل لهذه المصادر.
2. بعض وحدات الحاسوب لا يمكن أن تكون مشتركة من قبل أكثر من عمل ، وخوفا من مشكلة السباق والصراع على هذه الوحدات لا بد من جدولة الأعمال بطريقة لا تسمح بوجود السباق أو الصراع.
3. نظرا لتعدد البرامج فان تكلفة تنفيذ الأعمال ترتفع وذلك لزيادة زمن التنفيذ وزمن الانتظار لذلك لا بد من جدولة الأعمال لتحقيق التوازن بين عمليات المعالجة الحسابية وعمليات الإدخال والإخراج ولتحديد وقت انتظار العمل بحيث لا يتجاوز هذا الوقت قيمة محددة.

## طرق جدولة الأعمال (Methods of scheduling)

الهدف من جدولة الأعمال هو : تقليل وقت الانتظار لكل عمل وتقليل معدل أوقات الانتظار ولكافة الأعمال . توجد ثلاثة طرق أو مبادئ في الجدولة هي:

## أولا : مبدأ الداخل أولا هو الخارج أولا (FIFO)

## مثال :

ليكن لديك 4 أعمال كالتالي :1,2,3,4 أو ترتيب وصولها الزمني على التوالي  
10.0,10.1,10.2,10.3 كما موضح في الجدول التالي والزمن مقدرا بالساعات أي  
6=6×0.1: دقائق.

Job No رقم العمل	Arrival time وقت الوصول	Execution time	Finishing time	Waiting time وقت الانتظار
1	10.0	10.0	10.2	؟
2	10.1	10.2	10.5	
3	10.2	10.5	10.6	
4	10.3	10.6	11.0	

المطلوب : احسب وقت الانتظار لكل عمل ثم احسب معدل وقت الانتظار لكافة الأعمال حسب المبدأ الأول  
ثم اذكر تسلسل هذه الاعمال في التنفيذ .

حيث أن:  $w=F-A$

W:waiting time

F=finishing time

A=Arrival time

$$W_1=10.2-10.0=0.2$$

$$W_2=10.5-10.1=0.4$$

$$W_3=10.6-10.2=0.4$$

$$W_4=11.0-10.3=0.7$$

لحساب معدل وقت الانتظار لكل الأعمال =مجموع أوقات الانتظار للأعمال اعدد الأعمال

$$W=w_1+w_2+w_3+w_4$$

$$=0.2+0.4+0.4=1.7$$

$$W=W/N$$

$$W=1.7/4$$

$$W=0.425$$

سيكون تسلسل الاعمال حسب هذا المبدأ كالتالي : الاول ثم الثاني ثم الثالث ثم الرابع .

### ثانياً: مبدأ تنفيذ اقصر عمل أولاً (SJF (Shortest Job first)

يتم ترتيب الأعمال حسب وقت التنفيذ اللازم منها بحيث يتم تنفيذ العمل الذي وصل أولاً، ومن ثم يبدأ اختيار الأعمال التي تلي تبعاً لقصر مدة التنفيذ

مثال : احسب وقت الانتظار لكل عمل ثم احسب وقت الانتظار لكافة الأعمال حسب المبدأ الثاني في جدول الأعمال.

Job No رقم العمل	Arrival time وقت الوصول	Execution time	Finishing time	Waiting time وقت الانتظار
1	10.0	10.0	10.2	؟
2	10.1	10.3	10.6	
3	10.2	10.2	10.3	
4	10.3	10.6	11.0	

لحساب وقت الانتظار لكل عمل

$$W=F-A$$

$$(Job) W_1=10.2-10.0=0.2$$

$$W_2=10.6-10.1=0.5$$

$$W_3=10.3-10.2=0.1$$

$$W_4=11.0-10.3 = 0.7$$

لحساب معدل وقت الانتظار لكل الأعمال:

$$W=w_1+w_2+w_3+w_4$$

$$=0.2+0.5+0.1+0.7=1.7$$

$$W=W/N=1.7/4 = 0.375$$

رقم العمل	1	2	3	4
زمن الانتهاء	10.2	10.6	10.3	11
زمن البدء في التنفيذ	10.0	10.3	10.2	10.6
مدة التنفيذ	0.2	0.3	0.1	0.4

في هذا الجدول تم طرح وقت الانتهاء-وقت بدء التنفيذ.

فيكون تسلسل الأعمال الأول ثم الثالث ثم الثاني ثم الرابع.

### ثالثا : المبدأ الثالث: التنفيذ وفق معرفة المستقبل (future knowledge)

في هذا المبدأ يتم جعل الحاسوب خاملا من دون عمل حتى زمن وصول آخر الأعمال. ومن ثم ترتب كافة الأعمال تتابعيا من العمل ذي الوقت الأقصر للتنفيذ حتى العمل ذي الوقت الأطول للتنفيذ.

المثال : احسب وقت الانتظار لكل عمل ومعدل وقت الانتظار كافة الأعمال حسب المبدأ الثالث في جدولة الأعمال.

Job No	Arrival time	State of execution	Finish time	Wait time
1	10.0	10.4	10.6	0.6
2	10.1	10.6	10.9	0.8
3	10.2	10.3	10.4	0.2
4	10.3	10.8	11.2	0.9

لو أردنا حساب وقت الانتظار لكل عمل:

$$\text{Job1 } w_1=F-A=10.6-10.0 = 0.6$$

$$W_2=10.9-10.1 = 0.8$$

$$W_3=10.4-10.2 = 0.2$$

$$W_4 = 11.2 - 10.3 = 0.9$$

لو أردنا حساب معدل أوقات الانتظار = مجموع أوقات الانتظار \ عدد الأعمال

$$W = \frac{w_1 + w_2 + w_3 + w_4}{4} = \frac{0.6 + 0.8 + 0.2 + 0.9}{4} = \frac{2.5}{4} = 0.625$$

رقم العمل	1	2	3	4
وقت الانتهاء	10.6	10.9	10.4	11.2
وقت بدء التنفيذ	10.4	10.6	10.3	10.8
مدة التنفيذ	0.2	0.3	0.1	0.4

تم طرح وقت الانتهاء - وقت التنفيذ ينتج مدة التنفيذ .

فيكون تسلسل الأعمال الثالث ثم الأول ثم الثاني ثم الرابع .

مدة فترة حمول الجهاز = 0.3

**ملاحظة 1:** معدل وقت الانتظار زاد في المبدأ الثالث عن السابقتين .

**ملاحظة 2:** مبادئ الجدولة السابقة تمت على فرض عدم قدرة الحاسوب على استيعاب أكثر من عمل واحد في آن واحد.

- 1. العملية :-** هي برنامج في مرحلة التنفيذ . يحتاج تنفيذ العملية إلى عدة موارد منها وقت إلـ cpu وذاكرة وملفات و أجهزة إدخال وإخراج . يتكون نظام التشغيل من مجموعة من العمليات قسم منها تنفذ برامج تشغيل النظام وقسم منها تنفذ برامج المستخدمين . ويمكن جعل كل هذه العمليات تنفذ في نفس الوقت وذلك بتبديل وحدة المعالجة المركزية بين العمليات أي جعلها تعمل بتعدد (multiplexed) مما يجعل نظام التشغيل أكثر كفاءة . وكل عملية تمر بدورة متبادلة بين العمل داخل أـلـ cpu والعمل في وحدات الإدخال والإخراج لكي يتم تنفيذها . والمسؤول عن إدارة هذه العمليات وكل ما يتعلق بها من إنشاء أو إلغاء أو جدولة أو تزامن أو اتصالات وغيرها هو نظام التشغيل .
- 2. الجدول :** ويسمى جدول المهام أو الجدول قصير المدى (short term scheduler) وهو وحدة برمجية تمت برمجتها بناء على خوارزميات معينة بحيث تقوم بعملية اتخاذ قرار بشأن العملية التي سيتم تنفيذها.
- 3. المرسل : ( dispatcher ) :** هو وحدة برمجية مسؤولة عن إعطاء التحكم بالمعالج للعملية التي اختارها جدول المهام بحيث يقوم بعملية تبديل محتوى العملية التي يتم إيقاف تنفيذها بالعملية التي ستنفذ بدلا منها أي ما يسمى بتبديل المحتوى ( context switch ) .

**-الفرق بين الجدول والمرسل:-**

Dispatcher	Cpu scheduler
<p>-وظيفته تكمن في الجزء العملي أي انه هو الذي يقوم فعليا بالتنفيذ وجلب العملية التي اختارها الجدول وإرسالها إلى أـلـ cpu لتنفيذها ويتضمن عمله عدة أمور هي :</p> <ol style="list-style-type: none"><li>1. التبديل بين العمليات والبرامج أي (context switch).</li><li>2. نقل البرامج من قلب النظام إلى المستخدم (from kernel mode to user mode).</li><li>3. القفز إلى المكان المناسب في برامج المستخدم لإعادة تشغيل البرامج.</li></ol>	<p>يقوم الجدول باختيار عملية من العمليات الجاهزة من الذاكرة أي من طابور الاستعداد وليتم إدخالها إلى أـلـ cpu . أي أن وظيفته تكمن في اتخاذ القرار.</p>

4. وقت المرسل (latency dispatcher) : هو الوقت الذي يستغرقه المرسل حتى يوقف تنفيذ عملية ما ويبدأ تنفيذ العملية الأخرى وكلاهما يجب أن يكون سريعاً لأنهما يستخدمان كثيراً ومع كل العمليات.

5. تعدد المهام: يعني وضع عدد من العمليات في الذاكرة الرئيسية لكون جاهزة للتنفيذ.

6. جدولة المهام: تعني الاختيار من بين العمليات في الذاكرة الرئيسية لتكون جاهزة لتنفيذها إلى نقل العملية من الطابور الجاهز إلى الطابور التنفيذ.

7. كتلة السيطرة على العملية (Process control Block): هي تركيب بيانات في نواة النظام يحتوي على المعلومات اللازمة لإدارة عملية ويختلف تنفيذها من نظام لآخر. وهي تشمل بشكل عام الحقول التالية:-

- اسم العملية
  - حالة العملية (State) (استعداد-انتظار-تنفيذ-انتهاء).
  - عداد البرنامج الذي يشير إلى عنوان الأمر القادم الذي يستفيد في هذه العملية.
  - مسجلات الـ CPU مثل مؤشر المكس Stack pointer ومسجلات الفهرسة.
  - معلومات جدولة CPU ومعلومات إدارة الذاكرة والمعلومات الحسابية للعملية ومعلومات عن حالة الإدخال والإخراج.
  - مؤشر يحدد موقع القائمة التي تحتوي على جميع العمليات ذات الحالة نفسها.
- ملاحظة 1: تقع هذه الكتلة (PCB) في منطقة الذاكرة المحمية من وصول المستخدم العادي لأنها تحوي معلومات حساسة ومهمة.

ملاحظة 2: جميع العمليات التي لها نفس الحالة توضع في قائمة واحدة تسمى باسم الحالة كقائمة الاستعداد وقائمة الانتظار وهكذا...

## حالات العمليات (Process state)

كل عملية من العمليات لا بد أن تمر بأكثر من حالة وقت تنفيذها . وهذه الحالات تدل على نشاطها في تلك اللحظة.

الحالات التي تمر بها أي عملية هي:

### 1.التجديد (New): أي حالة لم تستخدم من قبل أو لم يفعل استخدامها .

وهي وقت تعريف العملية ووقت السماح لها بالدخول إلى قائمة العمليات الموجودة في الذاكرة الرئيسية RAM ويتم ذلك بالضغط على البرنامج ضغطة مزدوجة وبالتالي تنتقل هذه الحالة من الحالة الخاملة إلى حالة التنشيط.

### 2.الاستعداد (Ready):

وهي العملية الجاهزة للتنفيذ والدخول إلى وحدة المعالجة المركزية ولن يسمح لها بالتنفيذ بسبب وجود عملية أخرى تنفذ في نفس الوقت.

### 3.التشغيل أو التنفيذ (Running):

وهي حالة العمليات والأوامر وقت التنفيذ في الـ cpu

### 4.الانتظار (waiting):

وهي حالة العملية عند انتظار حدوث أمر معين . مثلا انتظار إدخال بيانات من المستخدم او عملية طباعة. (تكون العملية قد تمت بشكل جزئي).

### 5.الانتهاء (terminated):

وهي حالة العملية عند الانتهاء وهي إما أن تكون العملية قد انتهت بشكل سليم أو حصل لها خطأ معين أدى إلى إنهاءها.

**ملاحظة :** في حالة الاستعداد يمكن أن تكون أكثر من عملية في حالة استعداد في نفس الوقت وكل هذه العمليات مستعدة للتنفيذ في أي وقت . وفي حالة التشغيل عملية واحدة فقط تنفذ في الـ cpu في وقت معين . وحالة الانتظار تشبه الاستعداد فممكن أن تكون أكثر من عملية في حالة انتظار لحدث معين أيا كان هذا الحدث في نفس الوقت.



## مفهوم الطابور (queue):

هو مكان لجمع المهام بشكل منظم ومرتب وله عدة أنواع هي:-

1. **طابور المهام (Job queue) :-** توجد فيه جميع العمليات الموجودة في النظام.
2. **الطابور الجاهز (ready queue):-** يحتوي على جميع المهام (العمليات) المستعدة للتنفيذ.
3. **طابور الجهاز أو الانتظار (device queue or wait queue) :-** أو طابور الإدخال والإخراج أو طابور الأدوات. وهو يحتوي على جميع المهام (العمليات) التي تنتظر مدخلات أو مخرجات .

## ملاحظة:-

عندما تدخل عملية إلى النظام فإنها تدخل في طابور المهام (Job queue) الذي يحتوي على جميع عمليات النظام، وعندما تصبح العملية جاهزة وتنتظر التنفيذ فإنها تنتقل إلى الطابور الجاهز (ready queue) ، إما إذا كانت العملية تنتظر عملية إدخال أو إخراج مثل التحميل من القرص الصلب أو كانت تخدم اتصال انترنت فإنها تنتقل إلى طابور الانتظار .

## أنواع المجدولات حسب فترة عمل كل منها:

1. **المجدولات طويلة المدى (Long term scheduler)**  
وهي تدير العمليات من بدايتها إلى حين دخولها لطور الاستعداد وتعمل على نقل العملية من حالة التجديد إلى حالة الاستعداد.
2. **المجدولات متوسطة المدى (medium term scheduler)**  
يقوم برنامجها بعملية التبديل أي انه يحول العمليات بشكل مؤقت من الذاكرة الرئيسية إلى الذاكرة الثانوية وذلك حسب أولوية العملية وما تحتاجه من مساحة على الذاكرة.
3. **المجدولات قصيرة المدى (short term scheduler)**  
هذا النوع من المجدولات هو الأهم حيث يحدد العملية المؤهلة للانتقال لطور التنفيذ ويعمل على نقل العملية من حالة الاستعداد إلى حالة التنفيذ أي إلى المعالج ، وكذلك من حالة التنفيذ إلى حالة الانتظار عبر مراسل متخصص باداء هذا العمل يسمى (dispatcher).

## معايير تقييم جدولة المهام (scheduling criteria)

وهي معايير تستخدم في عملية المقارنة بين الخوارزميات المختلفة بهدف المفاضلة بينها .  
تلك المعايير تشمل:-

1. **الاستغلال الأمثل للمعالج:-** ويعني جعل المعالج مشغولا أي يعمل قدر المستطاع.
2. **الإنتاجية:-** وتعني عدد العمليات التي يتم إكمالها في وحدة زمنية محددة . وما نرغبه هو أعلى إنتاجية ممكنة.
3. **الوقت الدوري:-** هو الوقت المستغرق لإتمام دورة حياة العملية كاملة منذ دخولها وحتى انتهائها.
4. **وقت الانتظار:-** هو الوقت الذي تنتظره العملية في الطابور الجاهز منتظرة الحصول على المعالج أي منتظرة تنفيذها.
5. **وقت الاستجابة:-** هو الوقت المستغرق منذ دخول العملية حتى أول استجابة لها .

## جدولة العمليات :-

تعني وضع خطة لترتيب دخول العمليات على المعالج بحيث تدخل عملية واحدة في كل مرة ويستغل معظم وقت المعالج ويقوم مجدول العمليات بترتيب دخول العمليات على المعالج.

## خوارزميات جدولة العمليات

1. **خوارزمية العملية الواصلة أولا تنفذ أولا (FCFS) (first come first served) :**  
تعتمد هذه الخوارزمية على فكرة الطابور ولا يمكن لعملية أن تنفذ قبل عملية وصلت قبلها أبدا  
من ميزات هذه الخوارزمية :  
سهولتها:  
ومن عيوبها:-
  - إنها تظلم العمليات القصيرة الزمن والتي ستضطر للانتظار حتى انتهاء جميع العمليات التي تسبقها.
  - لا تعطي أهمية للعمليات الطارئة أو ذات الأولوية العالية.
2. **خوارزمية العملية الأقصر زمنا تنفذ أولا (STCF) (shortest time complete first) :**

في هذه الخوارزمية يتم قياس العمليات حسب زمن تنفيذها ويتم تنفيذ العمليات الأقصر أولا  
وهنا تظلم العمليات الطويلة الزمن حيث إنها تنتظر حتى يتم تنفيذ جميع العمليات التي قبلها.

يوجد نوعان من هذه الخوارزمية:

#### أ- النوع القابل للإجهاض (preemptive)

أي بمجرد وصول عملية قصيرة زمن التنفيذ إلى طابور الانتظار فإن المعالج يترك العملية الحالية ويتوجه نحو العملية الأقصر زمنا.

#### ب- غير القابل للإجهاض (non-preemptive)

إذا وصلت إلى طابور الانتظار عملية قصيرة الزمن وتوجد حاليا عملية طويلة الزمن قيد التنفيذ فإن المعالج لا يقاطعها وينتظرها حتى تنتهي.

#### 3. خوارزمية العمليات ذات الأولوية الأعلى تنفذ أولا:-

في هذه الخوارزمية يتم توزيع درجة أولوية لكل عملية. وتقوم أـ cpu بتنفيذ العمليات ذات الأولوية الأعلى . وفي حالة تساوي عمليتين في الأولوية تنفذ العملية الواصلة أولا (FIFS) يوجد نوعان من الخوارزمية:-

#### 1. الاجهاضية (Preemptive):-

عندما تأتي عملية تقارن أولوية العملية الجديدة بأولوية العملية التي تنفذ حاليا إذا كانت أولوية العملية الجديدة أعلى يجب إجهاض العملية التي تنفذ على أـ cpu حاليا وتنفذ العملية الجديدة التي تحتوي على أولوية أعلى.

#### 2. غير الاجهاضية (non-preemptive):-

عندما تأتي عملية وتكون أولويتها أعلى من أولوية العملية التي تنفذ حاليا على أـ cpu لا يقوم المعالج بإجهاض العملية الحالية.

- غير الإجهاض:- لا يسمح بإخراج العملية الحالية للتنفيذ بأي حال من الأحوال.  
- إجهاضي:- بمجرد وصول عملية ذات أولوية أعلى إلى طابور الانتظار فإن المعالج يجمد عمله مع العملية الحالية وينفذ العملية الواصلة حديثا ذات الأولوية الأعلى.

ملاحظة:- من عيوب هذا النوع من الجدولة التجويع (starvation) وهو أن العملية التي تحتوي على أولوية قليلة قد لا تنفذ أبدا. وللتغلب على هذه المشكلة تقوم بزيادة الأولوية لها مع مرور الوقت.

#### 4. خوارزمية (Round Robin):-

وتسمى جولة روبن أي مشاركة الوقت وتقسيمه بين العمليات. حيث يتم تقسيم الوقت إلى أجزاء بسيطة وتتناوب العمليات على المعالج في فترات، والعملية التي لم تكتمل وانتهى الجزء المخصص لها من الوقت تعود لطابور الانتظار وتنتظر دورها من جديد حتى تكتمل تماما. وهنا تحصل كل عملية على وقت عادل من وقت المعالج.

## 5. خوارزمية طابور متعدد المستويات:-

في هذه الخوارزمية تكون أولوية التنفيذ للعملية ذات الأولوية الأعلى مع السماح لها بجزء من الوقت يكون قصيرا جدا. أي إنها تعتمد على الأولوية ومشاركة الوقت.

-----

## الاتصال بين العمليات

أنواع العمليات في نظام التشغيل حسب الاتصالات بينها:-

### 1. عملية مستقلة (Independent process) :-

وهي العملية المستقلة التي لا تتأثر أو تؤثر في تنفيذ عملية أخرى في النظام وإنما تعمل مستقلة بذاتها.

### 2. عملية متعاونة (cooperating process) :-

وهي العملية التي يمكن أن تؤثر أو تتأثر بتنفيذ عملية أخرى في النظام. وهذه العملية تستخدم نوعين من الاتصال:-

#### أ- تبادل الرسائل أو الرسائل العابرة (message passing)

حيث يتم التبادل بين العمليات عن طريق الاتصال مع بعضها دون وجود مكان مشترك لتخزين الرسائل ويكون حجم هذه الرسائل ثابت أو متغير.

#### ب- الذاكرة المشتركة (Shared memory)

حيث يوجد بين العمليات مكان ذاكرة مشترك بينها يتم وضع الملفات المشتركة بداخلها.

- يتم الاتصال بين العمليات عن طريق برنامجين منفصلين عن بعضهما هما :-

1. برنامج المرسل:- يقوم بإرسال المعلومات اللازمة لتنفيذ العملية التالية في الترتيب.

2. برنامج المستقبل:- يقوم باستقبال المعلومات والبيانات اللازمة لتنفيذ العملية من العمليات السابقة.

## الفرق بين نظام تبادل الرسائل ونظام الذاكرة المشتركة

تبادل الرسائل	الذاكرة المشتركة
<ol style="list-style-type: none"> <li>1. جيد للتطبيقات التي لا تحتاج إلى اتصال كثيف بين المعالجات.</li> <li>2. سهولة التطبيق ومنخفضة الكلفة.</li> <li>3. وسيلة الاتصال عبر خطوط الاتصال في الشبكة.</li> <li>4. من سلبياتها تكلفة تمرير الرسائل عالية جدا لأنها تتطلب عمليات إدخال وإخراج كثيرة.</li> </ol>	<ol style="list-style-type: none"> <li>1. من حيث الاداء جيد وفعال</li> <li>2. من حيث التطبيق سهولة التطبيق ولكنها مكلفة ماديا.</li> <li>3. وسيلة الاتصال عبارة عن ذاكرة موحدة لجميع العمليات .</li> <li>4. من سلبياتها يجب توفير أساليب حماية للذاكرة المشتركة وتنظيم الوصول إليها.</li> </ol>

## تبديل العمليات (Context switching)

عبارة عن تبديل المعالج من عملية إلى أخرى أو من تشعب إلى آخر. ومجدول آلـ cpu هو من يحدد متى يتم تنفيذ عملية التبديل بين عمليتين.

**البيئة (context)** يتم التعبير عنها في كتلة السيطرة على العملية (process control block) لكل عملية وتتضمن القيم الموجودة في سجلات المعالج وحالة العملية، ومعلومات إدارة الذاكرة وغيرها.

تتم آلية تبديل العمليات من خلال المراحل التالية:-

1. تأجيل إكمال عملية من العمليات وحفظ حالة المعالج لهذه العملية في مكان ما في الذاكرة وذلك عند حدوث قطع مثلا (interrupt)
2. جلب محتوى البيئة (context) للعملية اللاحقة من الذاكرة وحفظها في سجلات المعالج.
3. الرجوع إلى المكان الذي يؤشر عليه عداد البرنامج للعملية التي حصل عندها القطع لإكمال العملية.

## أنظمة الحواسيب متعددة المعالجات

لزيادة فعالية الحواسيب ومرونة ادائها وزيادة كفاءتها الحسابية والمنطقية تضاف مجموعة من المعالجات إلى الحاسوب.

ومن أساليب ربط هذه المعالجات :-

### 1. معالجات متباعدة (Separate Systems):-

تكون على شكل حواسيب مستقلة بذاتها، كل منها مكتمل التركيب من ذاكرة مستقلة وطرفيات وغيرها.

### 2. معالجة موجهة (coordinated systems):-

كل معالج في كمبيوتر خاص به يقوم بوظيفة خاصة ويرتبط بجهاز مركزي واحد.

### 3. معالجات السيد الخادم (Master /slave System):-

تخضع جميع المعالجات إلى سيطرة من قبل معالج واحد يعمل عمل السيد مع العبيد . وأي تنقل بين المعالجات يتم من خلاله.

### 4. المعالجات المتجانسة (homogeneous system):-

وفيه يكون كل معالج قادرا على الاتصال والتحم بالمعالجات الأخرى وبنفس الوقت تتصل المعالجات بعضها ببعض في شبكة عمل واحدة وبذلك يتصف عمل كل معالج باللامركزية.

## المشاكل المتعلقة بالمعالجات وطرق حلها

تظهر المشكلات نتيجة لتعدد المعالجات ونتيجة لجدولة المعالجات والعمليات زمنيا. ومن المشكلات:-

### 1. مشكلة السباق (Race condition)

تقع هذه المشكلة حينما تشترك عمليتان أو أكثر ضمنيا أو بشكل صريح بنفس المصدر مثلا: عمليتان تطلب طباعة سطر على نفس الطابعة في نفس الوقت. فقد تظهر النتيجة خليط مبعثر من العمليتين مطبوع على نفس السطر.

-أحد الحلول المقترحة لهذه المشكلة هو الربط المسبق للآلة الطابعة قبل حدوث الطلب من إحدى العمليتين. مثلا ربط العملية الأولى بالطابعة مسبقا وعند طلب الطابعة من العمليتين في نفس الوقت تجول العملية الثانية بعد الأولى. وبذلك تنتهي حالة السباق وتكون نتائج كل عملية مستقلة عن الأخرى.

أن هذه الميكانيكية تسمى الطلب وفك الارتباط: (request and release) وهي من واجبات برنامج ضابط التحكم.

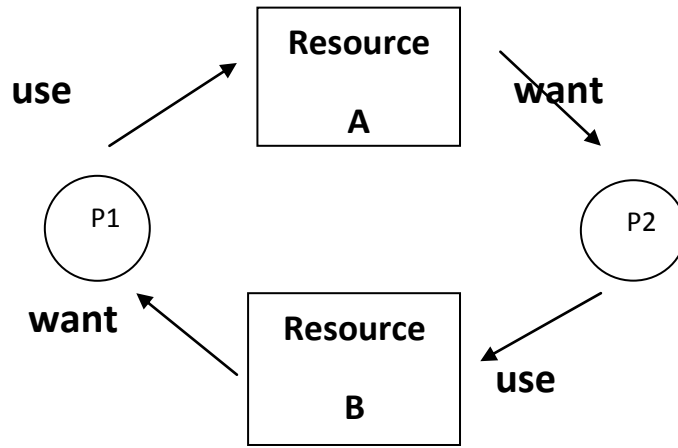
### 2. مشكلة الجمود (dead lock)

**الجمود:-** هو منع العمليات من استخدام موارد النظام بشكل دائم بسبب تنافسها مع العمليات الأخرى على هذه الموارد أثناء عملية الاتصال بينها.

- **مثال:-** شخصان يريدان رسم شكل هندسي احدهما يحمل قلم والأخر يحمل مسطرة فالذي يحمل قلم يطلب مسطرة والذي يحمل المسطرة يطلب القلم. كلا الطرفين لا يمكن تحقيقهما لان صاحب القلم لا يستغني عنه والثاني كذلك . فهنا يحدث الجمود.

## شروط حدوث الجمود

1. **منع التبادل:-** أي إذا كانت عملية واحدة يمكنها أن تستخدم المورد .
2. **الاستخدام والانتظار:-** أي إن تحجز العملية مورد معين وتطلب آخر تم إسناده لعملية أخرى.
3. **منع الإجهاض:-** أي إن العملية التي تحمل المورد لا يمكن إجهاضها ولا بد أن تقوم هي بتحرير الموارد التي تحملها.
4. **الانتظار الدائري:-** بمعنى أن تتكون دائرة مغلقة من الانتظارات تنشأ من احتكار عملية لمورد أو أكثر تنتظره عملية أخرى وهذه العملية بدورها تنتظر عملية أخرى وهكذا.....



## تجنب الجمود:- (Dead lock avoidance)

إذا طلبت العملية مصدر معين فلا يلبي الطلب إذا كان يمكن أن يؤدي إلى الجمود.

## منع الجمود: (Dead lock prevention)

يمنع الجمود بالتخلص من مسببات حدوثه ويعني ذلك :

1. مشاركة المصادر والسماح لها أن تستخدم في نفس الوقت.
2. إذا كانت العملية تمتلك مصدر فلا يمكن لها طلب مصدر آخر.
3. إذا كانت العملية تستخدم مصدر وتطلب آخر فعليها تحرير المصدر المحجوز ثم تطلب آخر.
4. ترتيب كافة المصادر في جدول والعمليات في جدول ومزامنتها.

**ملاحظة 1:-** يمكن أن يعمل نظام التشغيل لسنوات دون حدوث جمود فيه.



**ملاحظة 2:-** يتم كشف الجمود باستدعاء خوارزميات خاصة لهذا الغرض ويراقب الجمود بشكل دوري.

## معالجة الجمود:-

هناك خياران لكسر الجمود:

1. إجهاض المهام العالقة في الجمود.
2. منع المهام العالقة في الجمود من بعض المصادر.

### شرح النقطة الأولى:-

1. إجهاض المهام العالقة في الجمود يكون بطريقتين:-  
أ- إجهاض إحدى المهام التي تكسر حالة الجمود:

عندما تجهض إحدى المهام يقوم نظام التشغيل باستدعاء خوارزمية اكتشاف الجمود للتأكد من انتهائه أو وجوده . فإذا كان موجودا فلا بد من إجهاض مهمة أخرى وإعادة العمل حتى ينتهي الجمود.

### ب- إجهاض جميع المهام العالقة في الجمود:

وفي هذه الحالة سيحل الجمود ولكنها عملية مكلفة لان جميع المهام المجهضة تضطر لإعادة العمل وحجز المصادر وإضاعة الكثير من الوقت.

ملاحظة:- في الحالتين (أ،ب) يسترد نظام التشغيل جميع المصادر التي تمتلكها المهمة المجهضة.

### شرح النقطة الثانية:-

2. منع المهام العالقة في الجمود من بعض المصادر:

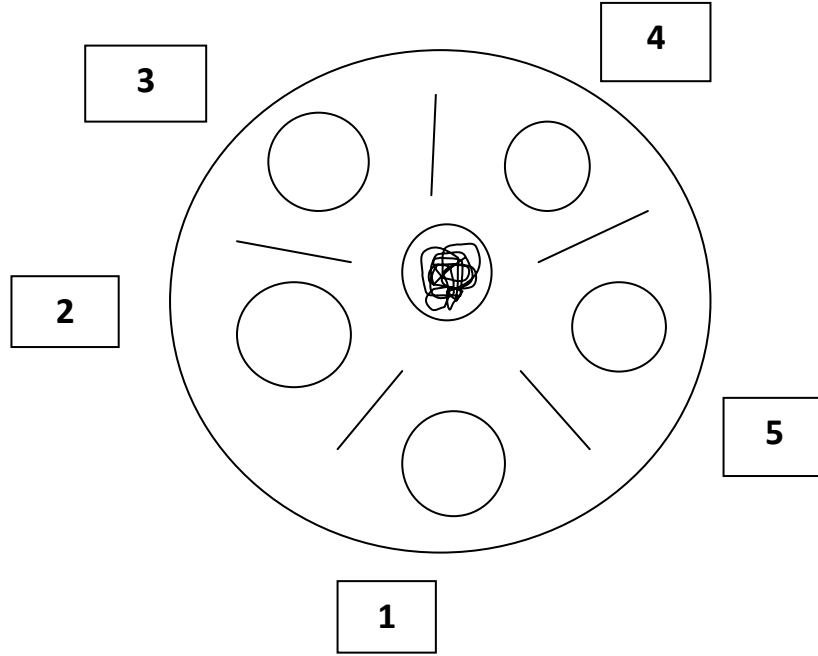
أي تحرم من بعض المصادر لكي لا تعلق في الجمود، وهناك ثلاث حالات هي:-

1. اختيار مهمة تكون ضحية لأخذ مصادرها.
2. إعادة تنفيذ المهمة بعد حرمانها من مصادرها.
3. التجويع (أي إن مهمة ما يجب ألا تكون ضحية دائما فقد لا تنتفذ أبدا ، ولتلافي ذلك نظام التشغيل يحسب عدد المرات التي صارت هذه المهمة ضحية).

## مثال على الجمود:- (طعام الفلاسفة)

لو كان خمسة فلاسفة يجلسون حول طاولة مستديرة وكل فيلسوف يقضي وقته ما بين التفكير والأكل وفي وسط الطاولة صحن كبير من السباغتي وسيحتاج الفيلسوف إلى شوكتين ليتناول حصته من السباغتي ولسبب ما.. لن يأخذ الفلاسفة سوى 5 شوك طعام ولذا ستوضع شوكة واحدة بين كل زوج من الفلاسفة وقد اتفقوا على إن كل منهما سيستخدمها عند حاجته لها بيده اليمين أو الشمال حسب موقع الشوكة منه.

وبعد بضع ثواني سيصبح كل من الفلاسفة جائعا وسيمسك كل منهم شوكة واحدة معه وسينتظر توفر شوكة ثانية له..



وبما انه لا يوجد المزيد من الشوك وكل منهم يحتاج لشوكتي طعام حتى يتمكن من الأكل فان أيا منهم لن يتمكن من الأكل وسيصلون لمأزق الجمود وعدم القدرة على المتابعة.

يقضي الفلاسفة حياتهم بين التناوب على التفكير والأكل. الفيلسوف بحاجة لالتقاط الشوك التي عن يمينه وعن شماله ليقوم بعملية الأكل ، عندما يبدأ الفيلسوف بالتفكير يضع الشوك على الطاولة مرة أخرى.

وخلال عملية الأكل يكون احد الفلاسفة يفكر أو جائع أو....

نجد مشكلة طعام الفلاسفة توفرت فيها شروط حدوث الجمود لان:-

1. فيلسوف واحد فقط يستطيع استخدام الشوكة في الوقت الواحد. مما يعني أن الشوكة مورد مقصود على فيلسوف واحد فقط.

2. الفيلسوف الجائع الذي معه شوكة واحدة فقط سوف يحتفظ بها ويحتجزها وسينتظر الحصول على شوكة أخرى.
3. بما إن الفلاسفة مسالمون ومهذبون فلا احد منهم سيرغب أو يحاول انتزاع الشوكة من جاره بالإجبار.

**التزامن :** هو أسلوب يستخدم في الانظمة متعددة المهام لتنظيم وصول العمليات الى موارد الحاسوب وتنفيذها للتعليمات بحيث لا يتعارض العمل بينها وتوجد انظمة مرور تستخدم لتحقيق التزامن مثل السيمافور ( semaphore ) .

**السيمافور:** هو منظم دخول العمليات على المعالج بحيث يضمن عدم دخول اية عملية في المعالج عند انشغاله بتنفيذ عملية اخرى وله حالتان :

1 – مفتوح ( open ) : يكون مفتوحا عندما يكون المعالج غير مشغول .

2 – مغلق ( locked ) : يكون مغلقا في حالة انشغال المعالج .

**ميكانيكية التزامن :** لكل مصدر من المصادر كوحدات الادخال والايخارج المشتركة او قاعدة البيانات يتوفر سيمافور خاص به ، فاذا كانت قيمة السيمافور مساوية للصفر فان المصدر يعتبر متوفرا ويمكن حجزه لتنفيذ عملية معينة . اما اذا كانت قيمته مساوية للواحد فان هذا يعني ان المصدر محجوز من قبل عملية اخرى (مغلق ) وقبل البدء باستخدام مصدر مشترك يجب :

اولا : فحص قيمة السيمافور المخصصة للمصدر المطلوب هل هي صفر ام واحد .

ثانيا : اذا كانت قيمته صفر اجعلها واحد ونفذ العمل وبعد الانتهاء من استخدام المصدر اجعل قيمة السيمافور تساوي صفر .

ثالثا : اذا كانت قيمة السيمافور واحد فهذا يعني ان المصدر محجوز ويجب الانتظار لحين انتهاء العمل الحالي .

## التشعبات (الخيوط) :

وهي طريقة تقسيم المهمة ( البرنامج ) الى عدة مهام لتسهيل وتسريع معالجتها من قبل نظام التشغيل بحيث يقوم البرنامج بتنفيذ هذه المهام بشكل متزامن .

**التشعب (الخيوط) thread :** هو سلسلة من التعليمات (الخطوات) في صورة روتين فرعي لاداء وظيفة معينة . ويمكن ان تحتوي المهمة على خيط واحد او اكثر، ولكل خيط :

- 1 – رقم تعريفى ( identification number ) .
- 2 – مسجل ( register ) .
- 3 – عداد ( program counter ) .
- 4 – مكسدس ( stack ) .

## تصنيف العمليات حسب التشعبات

- 1 – عمليات مفردة المهام ( single- threaded ) : تحتوي على تشعب واحد (خيط واحد) .
- 2 – عمليات سلاسل المهام المتعددة ( multi - threaded ) : اي يمكن تنفيذ سلسلة مهام بشكل متوازي على عدة انظمة حاسوبية . عمل التشعبات مع المعالجات المتعددة يكون اكفاً وافضل من المعالجات الوحيدة بسبب توزيع المهام بين المعالجات .

**ملاحظة 1:** الخيط يشكل مساراً لتنفيذ المهمة فهو لا يحتاج الى موارد خاصة به بل يستخدم موارد المهمة نفسها .

**ملاحظة 2 :** تجعل الخيوط البرنامج الحاسوبي يبدو وكأنه يقوم بأكثر من مهمة بشكل متزامن اذ يمكن تنفيذ تشعبات متعددة بشكل متوازي على نفس المعالج وهذا يحدث عن طريق تعدد المهام او مايسمى بتجزئة الوقت حيث يقوم ال-cpu بالتبديل بين التشعبات المختلفة ، وهذا

تزامن مزيف او كاذب . ويمكن انجاز تزامن حقيقي عن طريق الاستعانة بحاسوب متعدد المعالجات او معالجات متعددة الانوية .

## الفرق بين الخيط والمهمة

- 1 - الخيط هو جزء من المهمة .
- 2 - لكل مهمة عنوان مختلف وبيئة عمل مختلفة ومدة تشغيل معينة ، اما الخيوط فطالما انها مشتقة من المهمة فلها العنوان نفسه .
- 3 - الخيوط هي اجزاء متزامنة التنفيذ داخل مهمة ما .
- 4 - لا يوجد خيط من دون مهمة ولكن يمكن ان تكون مهمة بغير خيط .
- 5 - العملية مستقلة بذاتها بينما الخيوط ليست مستقلة .

## مميزات استخدام التشعبات المتعددة :

- 1 - رفع مستوى الاستجابة للمستخدم (responsiveness) .  
ان انشاء فروع للمهمة يرفع من مستوى الاستجابة للمستخدم مثلا يمكن للمستخدم اثناء استخدام احد برامج معالجة النصوص يعطي امر طباعة وفي نفس الوقت يمكنه طلب تدقيق املائي .  
ومثال آخر : يمكن لمستخدم الانترنت اثناء تحميل صفحة من الانترنت يمكنه ان يسجل الدخول الى بريده الالكتروني بينما الصفحة الرئيسية لازالت تقوم بتحميل بعض الصور .
- 2 - تقاسم الموارد ( resource sharing ) .  
عند انشاء فروع للمهمة فانها تشترك في موارد النظام مثل الذاكرة مما يساعد على توفير هذه الموارد ، اي تقليل المساحة المحجوزة للمهمة الواحدة ، اذ يمكن لجميع الفروع اجراء نشاطاتها المختلفة في نفس الحيز المحجوز . ولولا هذه الخاصية لاضطر نظام التشغيل الى اعطاء عنوان ومكان في الذاكرة لكل مهمة فرعية مما يسبب هدر في موارد النظام .
- 3 - الاقتصاد :  
ان انشاء مهمة يستغرق وقت من النظام لانه يجب ان يضمن نظام التشغيل عدم اختراق حدودها من قبل مهمة اخرى وكذلك عملية التبديل بين المهام المختلفة لرفع مستوى التزامنية في الاداء يضيف حملا على كاهل النظام ، اما عند انشاء التفرعات لا يحتاج نظام التشغيل الى احترازاات شديدة لان الفروع جميعها تشترك في نفس مساحة الذاكرة ، بالاضافة الى ان التبديل يكون اسرع بحوالي خمس مرات من التبديل بين المهام ، وانشاء المهام ابطأ بثلاثين مرة من انشاء التفرعات المتعددة .
- 4 - الاستغلال الامثل للنظام في حالة وجود اكثر من cpu .

تعدد المعالجات في النظام يرفع مستوى اداء الجهاز بشكل عام ، حيث يمكنه تنفيذ اكثر من مهمة في نفس الوقت مما يساعد على انجاز الاعمال بشكل اسرع لان المهام تتوزع على المعالجات فيكون استغلال كامل لموارد النظام .

---

## إدارة الذاكرة

# Memory Management

**ادارة الذاكرة :** تعني كيفية استغلال الذاكرة في جهاز الحاسوب الاستغلال الامثل بحيث يمكن الاستفادة من كل جزء من هذه الذاكرة دون هدر او تعطيل لهذه الاجزاء وبذلك يمكن تشغيل اكبر عدد من البرامج في آن واحد .

### **برامجيات ادارة الذاكرة :**

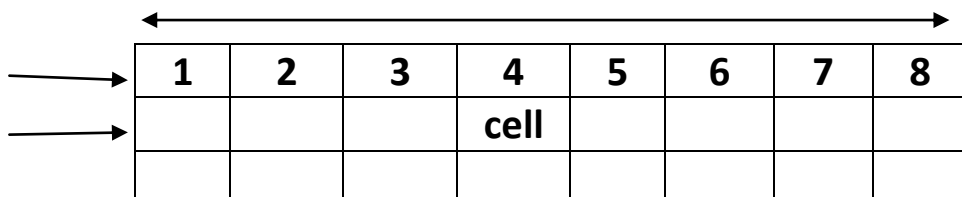
تعمل هذه البرامجيات على ادارة الذاكرة الرئيسية وتحميل البرامج ( البرنامج ) القابلة للتنفيذ وحفظها مؤقتا حتى تتم عملية التنفيذ .

**ملاحظة 1 :** يطلق اصطلاح المهمة او العمل ( job ) على البرنامج .

**ملاحظة 2 :** لتنفيذ أي برنامج يجب ان يحمل للذاكرة الرئيسية اولاً .

**ملاحظة 3 :** تتكون الذاكرة من مجموعة من الخلايا الثنائية موزعة بطريقة تشبه المصفوفة حيث يحدد عدد الخلايا في السطر الواحد طول الكلمة . يحمل كل سطر عنوان ( رقم ) وحيد لا يتكرر تستطيع الـ cpu من خلاله تحديد الكلمة المطلوبة في الذاكرة .

**طول الكلمة ( 8 bits )**



( Cell value either 0 or 1 )

( شكل تخطيطي لذاكرة )

## مهام برامجيات ادارة الذاكرة

- 1 - متابعة كافة مواقع الذاكرة وكافة الاعمال الداخلة اليها والمستقرة فيها والخارجة منها .
- 2 - رسم سياسة لمتابعة الذاكرة وذلك لتحديد المساحات التي يمكن اشغالها ومتى يسمح للاعمال بالدخول للذاكرة او الخروج منها .
- 3 - توطين الاعمال القابلة للتنفيذ في الذاكرة .
- 4 - تحرير مواقع الذاكرة المخصصة للاعمال التي تم تنفيذها وذلك بترحيل هذه الاعمال الى مصادرها الاساسية كالاقرص المغناطيسية .
- 5 - تقليل حجم الاجزاء غير المستغلة من الذاكرة ومحاولة الاستفادة منها .



# طرق ادارة الذاكرة

## اولا : ادارة الذاكرة المفردة المتماسكة

### (Single Contiguous Memory Management)

تعتبر هذه الطريقة من اسهل الطرق في ادارة الذاكرة . يرتبط مفهوم هذه الادارة بالحواسيب المصغرة ( الشخصية ) التي لاتستطيع في الفترة الزمنية الواحدة استقبال اكثر من عمل .

#### خصائصها الاساسية :

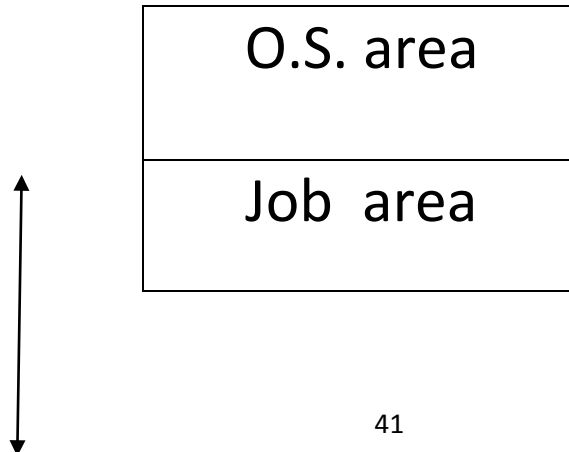
1 - تستوعب الذاكرة عملا واحدا فقط للتنفيذ .

2 - يجب ان يخزن العمل في مواقع متتالية ومتسلسلة ، اي يجب ان يكون العمل متماسكا ولا يجوز بعثرته في مواقع مختلفة .

تقسم الذاكرة الرئيسية في هذه الادارة الى قسمين اساسيين :-

1 - قسم يخصص لاستقبال العمل المراد تنفيذه يسمى منطقة المستخدم ( user area ) .  
بعض الاحيان تنشأ منطقة غير مستخدمة وذلك عندما يكون حجم العمل اقل من منطقة المستخدم .

2 - قسم يخصص لتخزين تعليمات نظام التشغيل .



## Unused Area

### تقسيمات الذاكرة المفردة المتماسكة

#### ملاحظة :

لغرض توفير حماية لمنطقة نظام التشغيل يوجد مسجل خاص يسمى مسجل الحماية ( protection register ) تخزن فيه قيمة آخر عنوان في منطقة نظام التشغيل ، وعند تنفيذ اي امر او ايعاز في البرنامج المنفذ يجب الا يتضمن عنوان تكون قيمته اقل او تساوي القيمة المخزونة في المسجل لان هذا يعني محاولة الوصول الى عنوان في منطقة نظام التشغيل حينها يتم رفض هذا الامر .

#### مزايا هذه الذاكرة :

يتميز هذا النوع من ادارة الذاكرة بسهولة الخوارزمية وقلة المعدات المستخدمة .

#### عيوب هذه الذاكرة :

- 1 – عدم استغلال الذاكرة استغلالا جيدا وذلك للأسباب التالية :
  - امكانية وجود منطقة غير مستخدمة اطلاقا .
  - امكانية احتواء البرنامج على تعليمات وبيانات تستخدم بشكل نادر ، الامر الذي يتطلب تواجدها مع البرنامج لعدم التمكن من تجزئة البرنامج ، لان هذه الادارة تتطلب وجود البرنامج كاملا في الذاكرة .
- 2 – تقييد حجم البرنامج المراد تنفيذه بحجم الذاكرة المتوفرة .
- 3 – عدم استغلال وحدة المعالجة المركزية جيدا وذلك لاحتمال انتظارها انهاء عمليات الادخال والاخراج المرتبطة بالبرنامج .

## خوارزمية حجز المواقع وتحريرها

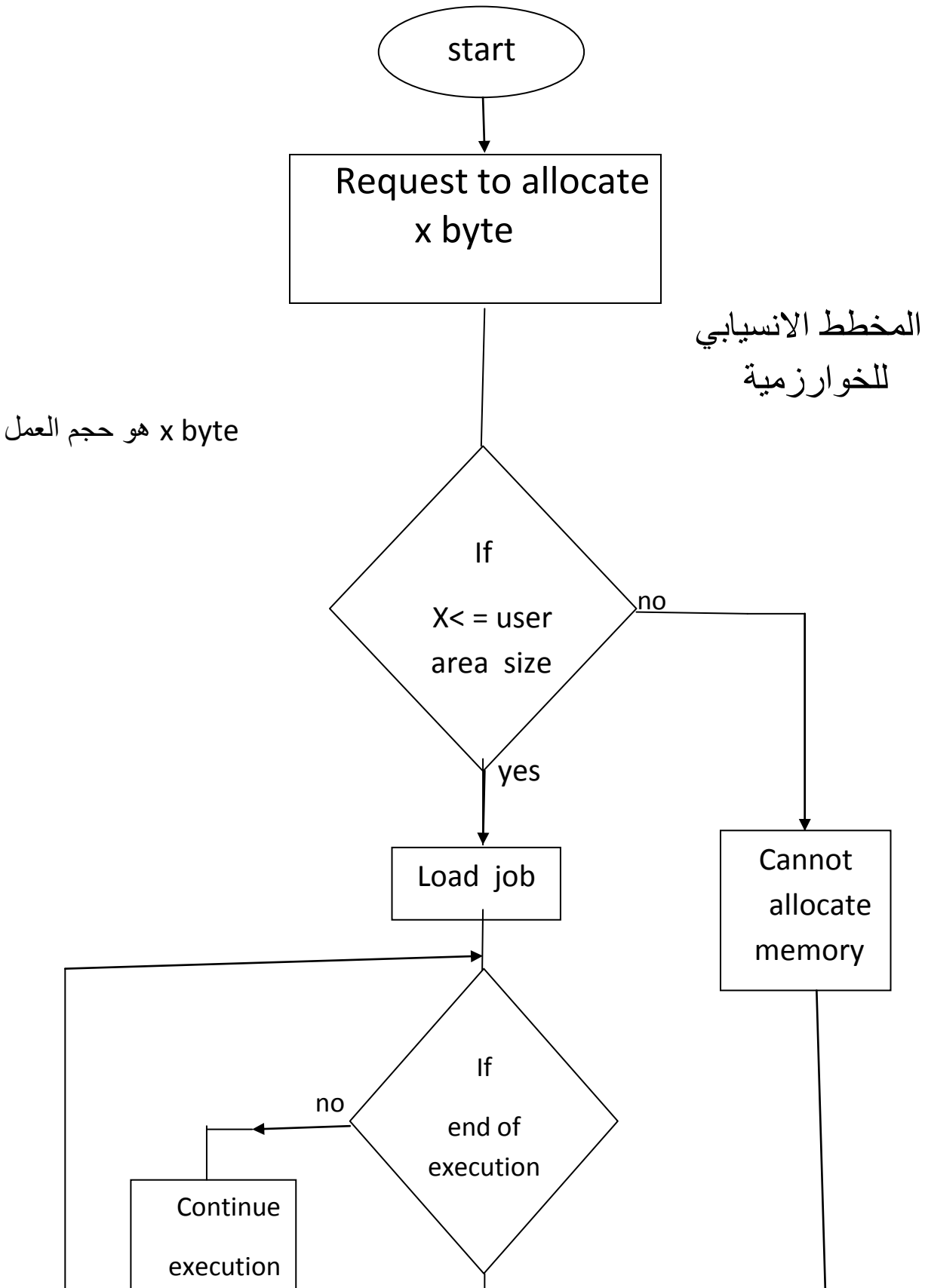
### فكرة الخوارزمية : ( algorithm idea )

1 - يتم تحديد حجم العمل المراد تنفيذه .

2 - يقارن حجم العمل بحجم منطقة المستخدم ، فاذا كان حجم العمل اقل او يساوي حجم منطقة المستخدم يحمل العمل في الذاكرة ويبقى متمركزا فيها حتى تنتهي عملية تنفيذه والا يهمل العمل ولا يحمل للذاكرة .

### THE ALGORITHM :

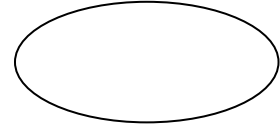
- 1 – request to allocate x byte
- 2 – if  $x \leq$  user area size then 5
- 3 – "cannot allocate memory
- 4 – goto 10
- 5 – load job
- 6 – if end of execution then 9
- 7 – continue execution
- 8 – goto 6
- 9 – release memory



yes

### شرح رموز المخطط الانسيابي :

يعني بداية ونهاية المخطط



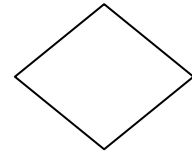
الرمز

يعني اجراء معالجات وعمليات ومعادلات



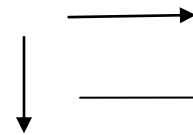
الرمز

يعني سؤال فيه اتخاذ قرار اما صائب او خاطئ ( yes or no )



الرمز

تعني اتجاه انسياب المعلومات



الرموز

## تعدد البرامج ( multi programming ) :

الفائدة من تعدد البرامج في اختصار وقت ال-cpu وبالتالي اختصار وقت المعالجة وذلك من خلال تنفيذ العمليات الحسابية والمنطقية وعمليات الادخال والاخراج على التوازي .

تختلف فكرة تعدد البرامج عن فكرة المشاركة الزمنية في ان مبدأ المشاركة الزمنية يتقبل برنامج واحد في الذاكرة الرئيسية في حين تبقى البرامج الاخرى في الذاكرة الثانوية حيث يعطى كل برنامج فترة (شريحة) زمنية للتنفيذ وعند انتهاء هذه الشريحة يدخل البرنامج الى الذاكرة الثانوية حتى عند اكمال معالجته لاحضار برنامج آخر حيث يعطى نفس الشريحة الزمنية للتنفيذ ، وهكذا .

## الفرق بين تعدد البرامج والمشاركة الزمنية :

### المشاركة الزمنية

### تعدد البرامج

1 - تحتاج حجم ذاكرة اقل .

1 - تحتاج لحجم كبير من الذاكرة لاستيعاب كافة

البرامج القابلة للتنفيذ .

2 - جدول عملية التنفيذ حسب عمليات الادخال

والاخراج .

2 - جدول عملية التنفيذ حسب الشريحة

الزمنية .

3 - لاحتياج البرامج الى عملية مبادلة لان كافة

البرامج موجودة في الذاكرة .

3 - تحتاج البرامج الى عمليات مبادلة بين

الذاكرة الرئيسية والثانوية .

4 - يفضل استخدامها في البرامج التي تكثر فيها

عمليات الادخال والاخراج مثل انظمة الحجز .

4 - يفضل استخدامها في البرامج التي

تكثر فيها العمليات الحسابية .

### **ثانيا :ادارة الذاكرة بالتجزئة الثابتة ( fixed partitioned memory management ) :**

تستخدم هذه الادارة فكرة تعدد البرامج وذلك من خلال تقسيم الذاكرة الى اجزاء ( partitions ) ثابتة الحجم قبل التنفيذ بحيث تختلف الاجزاء بعضها عن البعض الاخر في الحجم وذلك لاتاحة الفرصة لتحميل برامج باحجام مختلفة اليها .

وعادة مايكون عدد الاجزاء ثابتا ويحدد عدد الاجزاء درجة تعدد البرامج .

يتم في هذه الذاكرة تخصيص مسجلين لكل جزء تسمى مسجلات الحدود ( boundary register ) وذلك لحماية الاعمال والمحافظة على عدم تداخلها ، حيث يخزن في المسجل الاول عنوان البداية للجزء وفي المسجل الثاني يخزن عنوان النهاية لنفس الجزء .

عند تنفيذ تعليمة لبرنامج معين يتم مقارنة العنوان الذي ترتبط به هذه التعليمة بالقيم المخزونة في المسجل الاول والثاني للتأكد من ان العنوان يقع في المدى المخصص للجزء الذي يحتفظ بالبرنامج المحدد .

### **مثال :**

ذاكرة تتيح الفرصة لتحميل 5 اعمال في نفس الوقت وطلب منها تنفيذ الاعمال التالية :

Job 1 size = 40

Job 2 size = 10

Job 3 size = 200

### **الحل :**

تتم متابعة الذاكرة وتنفيذ عمليات التوطين والترحيل باستخدام جدول الحجز ( allocation table ) الذي يحتوي على مجموعة من الحقول : رقم الجزء ، حجم الجزء ، موقع الجزء ، حالة الجزء .

نفترض في هذا الوقت كانت هذه الاجزاء باجمعها فارغة ( free ) .

Part no.	Part size	Location	Status
1	20	360	Free
2	35	380	Free
3	85	514	Free
4	90	500	Free
5	434	590	Free

### جدول الحجز والتوطين في الذاكرة

استنادا الى جدول الحجز والتوطين يتم :

تحميل العمل الاول في الجزء الثالث

تحميل العمل الثاني في الجزء الاول

تحميل العمل الثالث في الجزء الخامس

ويكون شكل الذاكرة الرئيسية كالآتي :

RAM



O . S .
Job 2
Free
Job 1
free
job 3

ملاحظة : بقي فراغين (مساحتين فارغة) لتوطين عمليتين آخرين في هذه الذاكرة .

لاحظ بعد عملية الحجز والتوطين للاعمال الثلاثة السابقة ان الجزء الاول يحتوي على فراغ غير مستغل حجمه (10 k) والجزء الثالث يحتوي على فراغ حجمه (45 k) والجزء الخامس يحتوي على فراغ حجمه (234k) ، تسمى هذه الفراغات بالفراغات الداخلية ( internal fragment ) .

الذاكرة بعد عملية التوطين

O.S.
Job 2
10 unused
Free
Job 1
45k unused
Free
Job 3
234k unused

**مثال :** لو افترضنا وجود عمل بحجم 250k يحتاج للتنفيذ في الذاكرة السابقة فانه لن ينفذ بالرغم من وجود فراغات ومساحات حرة حجمها مجموعة اكبر من حجم العمل المطلوب . وهذه مساوئ هذا النوع من الذاكرة .

**خوارزمية الحجز والتوطين في الذاكرة باستخدام التجزئة الثابتة :**

**the algorithm**

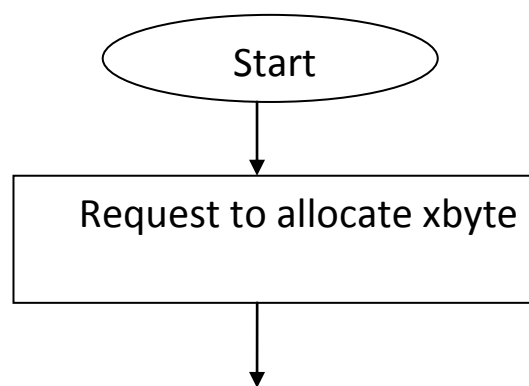
- 1 – request to allocate job with x byte
- 2 –  $i = 1$
- 3 – if  $x \leq \text{part size}(i)$  and  $\text{status}(i) = \text{"free"}$  then 5
- 4 – go to 9
- 5 – allocate this job in this part
- 6 – change the status (i) to "used"
- 7 – get new job
- 8 – go to 1
- 9 –  $i = i + 1$
- 10 – if end of table then 12
- 11 – go to 3
- 12 – end

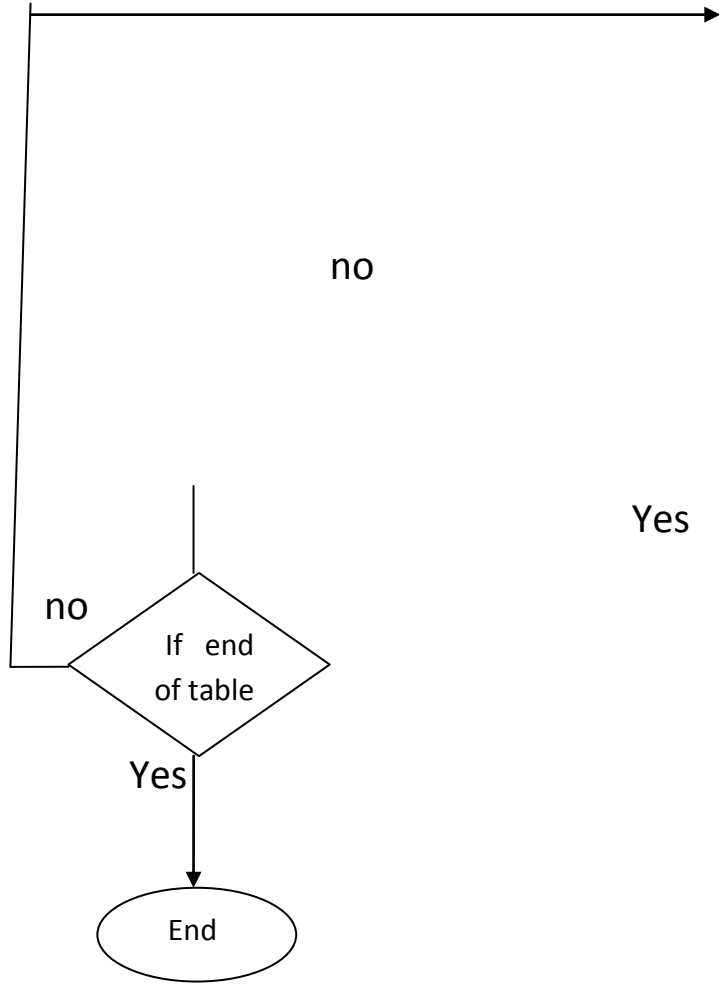
حيث ان : xbyte هو حجم العمل

: p.size هو حجم الجزء ( part size )

: i هو عداد يمثل رقم الجزء

المخطط الانسيابي لخوارزمية الحجز والتوطين باستخدام التجزئة الثابتة :





حيث ان p.size : مختصر  
 ( Part size ) أي حجم الجزء  
 i هو رقم الجزء (عداد )

### خوارزمية ترحيل الاعمال من الذاكرة باستخدام التجزئة الثابتة :

بعد الانتهاء من تنفيذ العمل يرحل من الذاكرة حسب الخوارزمية التالية :

- 1 - حدد رقم الجزء المطلوب تحريره .
- 2 - انتقل الى رقم الجزء المطلوب في جدول الحجز والتوطين وعدل حالته الى " free " .

3 – اذا انتهى الجدول ولم يصادف الرقم المطلوب فهذا يعني وجود خطأ في رقم الجزء المراد تحريره .

### مزايا هذه الذاكرة :

زيادة تعدد البرامج وذلك باتاحة الفرصة لتحميل اكثر من عمل قابل للتنفيذ .

### عيوب هذه الذاكرة :

1. وجود فراغات داخلية تؤدي الى ضياع وهدر جزء من الذاكرة وهدر وقت الـ cpu في استعراض مواقع الجزء كاملة بما فيها الفراغات .
2. تتطلب عملية التوطين والترحيل وقتا لاستعراض جدول التوطين واجراء عمليات التحديث اللازمة .
3. تتطلب عملية التنفيذ اجراء مقارنة للعناوين في الجزء مع القيم المخزونة في مسجلي الحدود الخاصة بالجزء .
4. بالرغم من تعدد البرامج الا ان حجم العمل لازال مقيدا بحجم اكبر جزء متوفر وكذلك ضرورة تماسك العمل في الجزء ، وضرورة تحميله بالكامل في الجزء المعني .
5. عملية التوطين تعتمد على اول مطابقة ، فلو كان الجزء الاول حجمه 200kb وكانت حالته حرة ، وطلب توطين عمل بحجم 10kb فهذا يعني توليد 190kb غير مستغل .

### ثالثا : ادارة الذاكرة بالتجزئة المتغيرة ( الحيوية)

#### **(Dynamic Partitioned Memory Management)**

في هذه الادارة تقسم الذاكرة الى اجزاء مختلفة الحجم اثناء عملية التنفيذ . ويتم تقسيم الذاكرة تبعا لحجم العمل القادم للتنفيذ فيصبح الحجم المشغول بالعمل جزءا ، وبهذا يتم الغاء الفراغات الداخلية .

تتطلب هذه الادارة توفر مجموعة من مسجلات الحدود حيث يحدد مسجلان لكل جزء كما في ادارة الذاكرة بالتجزئة الثابتة الا ان قيم المسجلات قد تتغير نظرا لتغير موقع وحجم الاجزاء .

بالاضافة الى المسجلات فان عملية الادارة تتطلب توفر جدولين هما :

1 - جدول المساحات المحجوزة ( allocated area table ) .

2 - جدول المساحات الحرة ( free area table ) .

تعتمد درجة تعدد البرامج على عدد البرامج التي يمكن تخزينها في الذاكرة ويكون محددا لايمكن تجاوزه .

**مزايا الذاكرة بالتجزئة المتغيرة :**

زيادة اداء وقدرة الحاسوب بزيادة درجة تعدد البرامج والتخلص من مشكلة الفراغات الداخلية .

**عيوب الذاكرة بالتجزئة المتغيرة :**

1. لازل حجم العمل مقيدا بحجم الجزء المتوفر ، ولتنفيذ العمل يجب ان يكون العمل مخزونا بالكامل في الجزء ومتماسكا .
2. هدر الوقت في معالجة الجداول ومتابعة مسجلات الحدود وذلك لتغير قيمها بين لحظة واخرى .
3. لازلت مشكلة الفراغات موجودة وتسمى في هذه الذاكرة الفراغات الخارجية ، وهي ناشئة عن وجود مساحات حرة موزعة لايسطيع ايا منها استيعاب عمل محدد .

**رابعا : ادارة الذاكرة بالتجزئة المتعددة :**

### **( Multiple Partition Memory Management )**

شرط تماسك العمل ليس ضروريا في هذه الذاكرة عكس الانواع السابقة ، وذلك لامكانية تقسيم العمل المراد تنفيذه الى اجزاء متفاوتة في الحجم شرط ان يكون حجم الجزء الواحد من مضاعفات الرقم 2 وذلك لتسهيل تحديد مواقع الاجزاء المختلفة في الذاكرة .

يشترط في ادارة الذاكرة بالتجزئة المتعددة توفر عدد من المساحات الحرة مناسبة في الحجم مساوية لعدد اجزاء العمل .

تتم معاملة الجزء الواحد في العمل كمعاملة العمل كاملا في الادارات السابقة .

### مزايا هذه الذاكرة :

انها تزيد من درجة تعدد البرامج وذلك بالغائها شرط تماسك العمل لتنفيذه وحل مشكلة الفراغات حلا جزئيا .

### عيوب هذه الذاكرة :

1. لازال العمل مقيدا بحجم الاجزاء المتوفرة في الذاكرة ولا زالت الذاكرة تتطلب توطين العمل كاملا فيها مما يؤدي الى عدم استغلال الذاكرة بشكل جيد وهدر في وقت الـ cpu في استعراض مواقع في الذاكرة غير مستخدمة او تستخدم بشكل قليل .
2. كثرة المعدات الفيزيائية (كالجداول والمسجلات المستخدمة والتي تحتاج الى حيز في الذاكرة لتخزينها ووقت من الـ cpu لمتابعتها من حيث الانشاء والتعديل والحذف .
3. لايسمح بترحيل العمل من الذاكرة الا بعد انتهاء تنفيذه كاملا مما يؤدي الى وجود اجزاء غير عاملة وغير مستغلة في الذاكرة .
4. احتمالية وجود فراغات داخلية في جزء العمل لان حجم الجزء من مضاعفات العدد 2 .

### مثال :

احجز موقع في الذاكرة لعمل بحجم (27kb) باستخدام ادارة الذاكرة بالتجزئة المتعددة :

الحل : يقسم العمل الى اجزاء مثلا : 2,8,8,8,2

أو : 2,4,6,8,8

وغيرها من التقسيمات . وكل التقسيمات يتولد منها فراغ يبقى مصاحبا للعمل عند توطينه في الذاكرة .

### خامسا : ادارة الذاكرة باستخدام عملية الضغط ( compaction ) :

للاستفادة من الاجزاء الحرة المتوفرة ولزيادة درجة تعدد البرامج ولتقليل الفراغات قدر الامكان تستخدم ادارة الذاكرة بالضغط أي اعادة توزيع المواقع . تتلخص خوارزمية هذه الادارة بما يأتي :

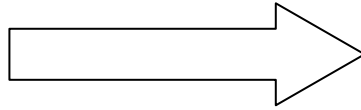
1. قبل تنفيذ العمل يحمل للذاكرة اذا توفر جزء يستوعب هذا العمل .
2. اذا لم يتوفر جزء حر فيتم حساب مجموع المساحات الحرة .
3. اذا كان حجم المساحات الحرة اكبر من او يساوي حجم العمل يتم تجميع الفراغات وذلك بضغط الذاكرة لانتاج منطقة حرة واحدة تستطيع استيعاب العمل المراد تنفيذه .

**مثال :** افترض لديك الذاكرة التالية وتطلب الامر تنفيذ عمل بحجم (40kb) باستخدام هذه الطريقة :

O.S.
Job1
Free
Job2
Free
Job3
Free

الذاكرة قبل عملية الضغط

الذاكرة بعد عملية الضغط



O.S.
Job1
Job2
Job3
Free area

في المنطقة الحرة (free area) .

الثاني تمت ازاحته بمقدار 32kb للاعلى . والعمل الثالث 40kb للاعلى .

الحل : يحمل العمل

ملاحظة 1 : العمل تمت ازاحته بمقدار

ملاحظة 2 :



عند ازاحة العمل في الذاكرة تبقى محتويات البرنامج ثابتة اي ان تعليمات البرنامج لا تتغير بحيث تبقى العناوين التي تتعامل معها التعليمات كما كانت قبل عملية الضغط ، وتسمى هذه العناوين العناوين المنطقية او الافتراضية ( virtual or logical addresses ).

اما العناوين الفيزيائية للاعمال ( physical addresses ) فقد اختلفت وذلك لحدوث تغيير على مواقع هذه الاعمال بعد عملية الضغط ، لذا لابد من تخصيص مسجل خاص لكل عمل يسمى مسجل اعادة ترتيب المواقع ( relocation register ) يخزن فيه مقدار الازاحة للعمل .

### مزايا هذه الذاكرة :

فيها امكانية تنفيذ عملية الضغط التي تؤدي الى زيادة عدد البرامج المحملة الى الذاكرة .

### عيوب هذه الذاكرة :

1. الحاجة الى توفر مسجلات خاصة وبالتالي استغلال مواقع في الذاكرة لحفظ قيم الازاحة .
2. هدر وقت وحدة المعالجة المركزية في تنفيذ عملية الضغط ، وفي احتساب قيم الازاحة لكل عمل ، وفي احتساب العنوان الفيزيائي عند تنفيذ كل تعليمة من تعليمات العمل .

**التفتت الداخلي :** هو فراغ ضائع في الذاكرة داخل كل جزء ينتج عن توطين مهام في الذاكرة حجمها اصغر من حجم ذلك الجزء .

**التفتت الخارجي :** هو فراغ ضائع في الذاكرة لا يتسع لاي مهمة لصغر حجم ذلك الجزء .

## سادسا : ادارة الذاكرة بالصفحات ( paged memory management ) :

في هذه الادارة يتم تقسيم الذاكرة الرئيسية ( ram ) الى اطارات خالية ( page frame ) .

هذه الاطارات تكون جاهزة لاستقبال مايساوي حجمها من اجزاء صفحات منطقية تمثل اجزاء البرنامج المراد تشغيله وبذلك يوضع كل جزء من البرنامج في اطار مناسب له في الذاكرة بحيث يكون حجم الاطار اكبر من او يساوي حجم جزء البرنامج الموضوع عليه .

### جدول الصفحات الذاكرية ( page table ) :

وهو جدول مواقع الذاكرة ، وهو مفهرس بارقام الصفحات . كل مدخل (entry) في هذا الجدول يعطي العنوان الفيزيائي المطابق للعنوان المنطقي .

### آلية عمل هذه الذاكرة :

1. يتم تقسيم العمل الى اجزاء متساوية في الحجم ، وبحجم من مضاعفات العدد 2 لتسهيل عملية العنوان . يسمى كل جزء بالصفحة ، والذاكرة كذلك يتم تقسيمها الى اجزاء متساوية في الحجم وبحجم الصفحة ، وكل جزء يسمى كتلة ( block ) .
2. تعتبر الصفحة وحدة فيزيائية للبيانات حيث يمكن القراءة منها والكتابة عليها . ويمكن ان تحتوي على فراغات داخلية خاصة اذا كان حجم العمل المراد تحميله ليس من مضاعفات العدد الرقم 2 .
3. لكي يتحمل العمل لابد من تحميله كاملا في الذاكرة ، اذا لابد من توفر عدد من الكتل مساوي او اكبر مساوي او اكبر من عدد صفحات العمل بحيث تكون هذه الكتل غير مشغولة .
4. ليس من الضروري تماسك العمل ، اذ يمكن تحميل صفحات العمل في كتل متباعدة .
5. ترحيل العمل يتم دفعة واحدة ، اي ترحل كافة صفحاته عند الانتهاء من تنفيذه .

**ملاحظة :** حجم العمل المراد تحميله يعتمد على حجم الذاكرة المتوفر .

**مزايا هذه الذاكرة :** زيادة عدد البرامج وتقليل الفراغات الخارجية قدر الامكان مما يؤدي الى تحسين عمل المعالج وتحسين استعمال الذاكرة .

### عيوب هذه الذاكرة :

1. لازال حجم العمل مقيدا بحجم الذاكرة ولا زالت عملية التنفيذ تتطلب تحميل العمل كاملا في الذاكرة .
2. وجود الفراغات الداخلية داخل الصفحة .
3. كثرة المعدات الفيزيائية المطلوبة مثل (جدول الاعمال ، جدول الكتل ، مسجلات العناوين ، ... )

### سابعا : ادارة الذاكرة الافتراضية ( virtual storage management ) :

ان المعالج لا يشترط وجود البرنامج كاملا في الذاكرة الرئيسية ليتمكن من معالجته .

فهو غالبا لا يحتاج كل اقسام البرنامج في نفس الوقت بل في اوقات مختلفة خلال عملية المعالجة ، لذا تم الاستعانة بتقنية الذاكرة الافتراضية لتجنب حجز مكان لا يستغل في الذاكرة .

المبدأ الأساسي لتقنية الذاكرة الافتراضية هو عدم تحميل جميع صفحات البرنامج الى الذاكرة الرئيسية ( الحقيقية ) مباشرة عند بداية تنفيذ البرنامج بل تحميلها حسب الحاجة اثناء التنفيذ .

### مزايا وفوائد تقنية الذاكرة الافتراضية :

1. يمكن للمبرمجين برمجة العديد من البرامج الطويلة بغض النظر عن محدودية الذاكرة الرئيسية .
2. يمكن العمل على عدة برامج في نفس الدين وان كانت الذاكرة الرئيسية لاتتحمل حجم هذه البرامج مجتمعة .
3. عندما يضع المبرمج برنامج في الذاكرة يتخيل ان هناك مساحة افتراضية كبيرة لبرنامجها بينما الواقع غير ذلك .
4. الذاكرة الافتراضية تعطي صورة للمستخدم بان برنامج تم تحميله كاملا في الذاكرة الرئيسية خلال فترة المعالجة ، وتعطي البرنامج التطبيقي الانطباع بان هناك ذاكرة متصلة يعمل عليها ، بينما في الحقيقة البرنامج مجزأ الى اجزاء . وعند تحميل البرنامج لمعالجته يتم تحميل بعض الاجزاء للذاكرة والبعض الاخر يتم وضعه في مساحة في القرص الصلب .

### عيوب هذه الذاكرة :

مكلفة من ناحية المساحة الدائمة (backing store) ومن ناحية الوقت لانها تزيد عدد المرات مقاطعة البرنامج (او البرامج )، وتحتاج الى برمجة معقدة .

### خطأ الصفحة (page fault) :

عندما يرغب المستخدم في تنفيذ تعليمة من برنامج موجودة في صفحة لم يتم تحميلها في الذاكرة سينتج عن ذلك خطأ الصفحات ، وهذا تقصير في عمل النظام في احضار صفحات البرنامج للذاكرة . ولمعالجة هذا الخطأ سيضطر المصفح (pager) لاحضار هذه الصفحة من القرص الى الذاكرة ، ويتم تسكينها في اطار خالي من الذاكرة ، ثم يعدل جدول الصفحات الذاكرية ليشتمل وجود هذه الصفحة ، ثم يتم اعادة العملية التي طلبت هذه الصفحة .

### استبدال الصفحات ( replacement )

عند وجود خطأ الصفحة وعند عدم وجود مكان فارغ في الذاكرة يختار النظام صفحة ضحية ليستبدالها بالصفحة المطلوبة . وعلى ذلك فان خوارزمية استبدال الصفحات هي التي تختار وتقرر الصفحة الخارجة من الذاكرة الى القرص الصلب .

### خوارزمية استبدال الصفحات :

1. البحث عن موقع الصفحة المطلوب جلبها من القرص وتحميلها الى الذاكرة .
2. البحث عن اطار فارغ في الذاكرة وله حالتان :

- اذا وجد اطار فارغ في الذاكرة يتم استخدامه .

- اذا لم يجد اطار فارغ يتبع احدى خوارزميات استبدال الصفحات لاختيار الاطار

الضحية .

3. يتم نقل محتويات الاطار الضحية الى القرص ويتم تعديل جدول الصفحات الذاكرة ليعكس عدم وجود هذه الصفحة في الذاكرة .
4. يتم احضار الصفحة المطلوبة الى الاطار الذي يتم تفريره من الذاكرة .
5. اعادة تنفيذ الامر الذي طلب هذه الصفحة .

-----

### ثامنا : المبادلة ( swapping ) :

وهي طريقة من طرق ادارة الذاكرة تتضمن ترحيل عمل او حزم من عمل من الذاكرة الرئيسية الى الذاكرة المساندة وبالعكس ، وتوطين العمل او جزء العمل في المكان المناسب من الذاكرة المساندة في نفس مواقع العمل او جزء العمل المرسل .

عملية المبادلة تنقسم الى عمليتين او مرحلتين :

1. مرحلة سحب العمل او جزء العمل من الذاكرة المساندة الى الذاكرة الرئيسية وتعرف باسم الدحرجة الى الذاكرة الاساسية (rolling in).
  2. مرحلة سحب العمل او جزء العمل من الذاكرة الاساسية الى الذاكرة المساندة ، وتعرف هذه العملية بالدحرجة خارج الذاكرة (rolling out) .
- تتم المرحلتان على نفس الموقع من الذاكرة الاساسية .

-----

# برامجيات ادارة المعلومات

## أنظمة الملفات

مفهوم برامجيات ادارة المعلومات :

تخزن المعلومات على وحدات التخزين الثانوية كالقرص المغناطيسي مثلا .

يتم تنظيم هذه المعلومات في ملفات يمكن الرجوع اليها عند الحاجة لتنفيذ عمليات المعالجة المختلفة عليها .

## ملف الحاسوب :

هو مجموعة من المعلومات يتم تخزينها في الحاسوب بصيغة معينة .

اي انه سلسلة متتالية من البايتات الثنائية وهذه البايتات قد تمثل اعداد صحيحة او احرف نصية او صورة او ملف صوتي .... الخ ، ويقدم للمستخدم بالشكل المفهوم له . يكون الملف متاحا للبرامج لاستخدامه بعد انتهاء البرنامج الحالي .

حجم الملف : هو عدد البايتات التي يتكون منها الملف اي المساحة التخزينية التي يشغلها الملف .

## مهام برامجيات ادارة المعلومات :

- تهتم برامجيات ادارة المعلومات بالهيكل التنظيمي ، اي التركيب الفيزيائي لهذه المعلومات والمتمثل في كيفية تخزين المعلومات على الوسط الفيزيائي ، وكيفية الرجوع اليها من خلال تحديد العنوان الفيزيائي لها . نظام الملفات يهتم بالبنية الفيزيائية للمعلومات ويقصد بهذا المفهوم :

- 1 - تهيئة البنية الفيزيائية وجعلها قادرة على استقبال المعلومات وذلك من خلال فحص المسارات المكونة لاسطوانة التخزين واستحداث مايلزم من قطاعات (sectors) مثل القطاعات التي تحدد مواقع الادلة وجدول حجز الملفات ( FAT ) .
- 2 - استحداث مناطق للدليل ، والدليل هو جدول يتكون من مجموعة من المداخل ، يخصص كل مدخل لملف واحد ويضم المدخل الواحد معلومات عن الملف مثل : اسم الملف ، امتداده ، حجمه ، قيوده ، تاريخ انشاؤه ..
- 3 - استحداث الملفات .

نظام الملفات : يعني طريقة تخزين وتنظيم وادارة ملفات الحاسوب والبيانات التي تحتوي عليها تلك الملفات لتسهيل ايجادها واستخدامها .

## وظائف نظام الملفات

- 1 – متابعة المعلومات ( الملفات ) المخزونة في الذاكرة الثانوية اعتمادا على دليل الملفات وجدول توزيع الملفات .
- 2 – تحديد سياسة متابعة الملفات على وسائط التخزين وكيفية الوصول الى الملفات ( تتابعيا او باستخدام فهرس او عشوائيا ) .
- 3 – تخزين الملفات على واسطة التخزين وترحيلها للذاكرة الرئيسية عند الحاجة لمعالجتها .
- 4 – تحديث المعلومات على واسطة التخزين والغاؤها اذا تطلب الامر .
- 5 – تحديد المساحة الحرة والمستخدمه من اجمالي مساحة القرص الصلب .
- 6 – حفظ او معرفة اسماء الادلة والملفات .
- 7 – معرفة او تحديد الموقع الفيزيائي للملف على القرص الصلب .

-

## مكونات نظام الملفات



1. نظام الملف الرمزي ( symbolic file system ) الذي يبحث في منطقة الدليل عن اسم الملف المطلوب .
2. نظام الملف الاساسي ( basic file system ) الذي يعمل على قراءة المعلومات الاساسية عن الملف في منطقة الدليل .
3. نظام حماية الملف ( protection file system ) يقوم هذا النظام بمطابقة قيود الملف مع شفرة العملية المراد تنفيذها على الملف للتأكد من ان العملية المراد تنفيذها مطابقة لمحتوى قيود الملف في دليل الملفات .
4. نظام الملف المنطقي ( logical file system ) الذي يتولى حساب السجل المنطقي في الملف.
5. نظام الملف الفيزيائي ( physical file system ) الذي يعمل على تحويل العنوان المنطقي للسجل في الملف الى عنوان فيزيائي للسجل على واسطة التخزين .
6. نظام التوطين ( strategic allocation system ) يعمل هذا النظام على قراءة السجل او كتابته وذلك بنقل السجل من منطقة التخزين المؤقتة في الذاكرة الرئيسية الى القرص في حالة الكتابة او نقل السجل من القرص المغناطيسي الى الذاكرة المؤقتة لمعالجته في حالة القراءة .

1. نظم ملفات القرص (disk file systems) : تستخدم لتخزين البيانات على القرص مثل :
  - نظم جداول توزيع الملفات ( FAT ) وهي على انواع مثل: FAT 12- FAT 16- FAT 32
  - ( new technology file system ) NTFS .
2. نظم ملفات الفلاش ( flash file systems ) : تستخدم لتنظيم الملفات على وحدة الذاكرة الوميضية .
3. نظم ملفات قواعد البيانات ( data base file systems ) : تستخدم لترتيب وتخزين بيانات قواعد البيانات .
4. نظم ملفات الشبكات ( network file systems ) : تستخدم لتسهيل الوصول للملفات واستخدامها عبر الشبكات .
5. نظم ملفات القرص المشترك ( shared disk file systems ) : في هذا النوع تشترك عدة حواسيب في استخدام قرص خارجي واحد ، يقوم نظام الملفات بمنع التداخل فيما بينهم عند كتابة البيانات على القرص .

## اولا: ( FAT 16 )

هو مختصر ( file allocation table ) ويعني جدول تخصيص الملفات .

### خصائصه :

1. يستخدم عناوين (16 bit) .
2. يعمل هذا النظام مع (ms-dos / version 4) ومع كل اصدارات ( windows ) .
3. يهدر يهدر هذا النظام مساحة كثيرة ، فكلما يحفظ المستخدم عددا كبيرا من الملفات الصغيرة كلما تراكم حجم السعة التخزينية المهدورة .

## ثانيا : ( FAT 32 )

### خصائصه :

1. يستخدم عناوين ( 32 bits ) .
2. يستخدم في انظمة التشغيل  
a. ( windows 95 – windows 98- windows me - windows xp - windows2000 ) .
3. المساحة التخزينية المهدورة اقل من نظام ( FAT 16 ) .
4. اكثر كفاءة من نظام ( FAT 16 ) ولكن يظهر قصوره مع نسخ ( windows ) القديمة حيث لا يمكنها التعرف على تقسيماته الا بمشغلات خاصة .

## ثالثا : ( NTFS )

هو مختصر ( NEW TECHNOLOGY FILE SYSTEM )

## خصائصه :

1. يستخدم في انظمة التشغيل ( windows 7- windows vista- windows xp- windows nt – windows me- windows 8 )
2. يوفر درجة عالية من الحماية والامان لوحدته الخزن مقارنة بنظام ( FAT 32 ) ، حيث يستطيع التحكم اكثر في حما الملفات والمجلدات وتحديد المستخدمين الذين يمتلكون صلاحية في فتح ملف او برنامج او مجلد ما .
3. يعتمد هذا النظام على جدول الملفات الرئيسي ( MASTER FILE Table ) حيث يقوم النظام ( NTFS ) بعمل نسخة احتياطية لهذا الملف الرئيسي لحماية البيانات من التلف ويقوم باستدعاء النسخة الاحتياطية في حالة عطل النظام وعمل نسخة احتياطية له لاحقا .
4. يفضل استخدامه للاقسام الكبيرة الحجم من القرص الصلب والتي تتجاوز ( 40 GB ) لانه يستخدم مساحة خزنية كبيرة لهيكله النظام .
5. يسمح هذه النظام بضغط الملفات او المجلدات وتصغير حجمها بشكل ملحوظ دون الحاجة الى ضغط القرص كاملا .
6. يتيح هذا النظام لمدرء الشبكات ان يسمحوا او يمنعوا لنفاد لتقسيمات محددة .
7. يمكن استعادة عدد اكبر من الملفات المفقودة .
8. يوفر امكانية تحديد سعة خزنية معينة لكل مستخدم على نفس وحدة الخزن ولا يسمح بان يتجاوز اي مستخدم على المساحة الخزنية لأي مستخدم آخر .

## عيوبه :

1. اذا كان المستخدم يستخدم نظامين ، كل نظام على قرص صلب فلن يستطيع مشاهدة ملفات ( NTFS ) على قرص آخر بسبب الحماية العالية .
2. نظام ( FAT 32 ) اسرع من ناحية التنقل والتحكم من نظام ( NTFS ) .

## معالجة الملفات

يقصد بمعالجة الملفات اجراء العمليات الاتية على الملف :

1. تهيئة مكان على الشريط او القرص المغناطيسي مخصص للملف ، وتسمى عملية التهيئة هذه بعملية تكوين الملف فتح الملف ( open file ) .

2. كتابة البيانات في الملف ويقصد بها عملية نقل البيانات من ذاكرة الحاسوب الرئيسية الى الذاكرة المساعدة .
3. بعد الانتهاء من ادخال البيانات في الملف لابد من غلق الملف ، حيث تسمى هذه العملية غلق الملف ( close file ) .
4. قراءة البيانات من الملف يقصد بها عملية نقل البيانات من الذاكرة المساعدة ( شريط او قرص مغناطيسي ) الى ذاكرة الحاسوب الرئيسية لغرض اجراء المعالجة عليها .
5. تحديث الملف ، يقصد بها حذف قيد من الملف او اضافة قيد جديد الى الملف او اجراء تعديلات على قيود موجودة سابقا .

-----